САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Цзоу Цзиньин

ОБЪЯСНИМЫЙ ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ДЛЯ СИСТЕМ БОЛЬШОЙ РАЗМЕРНОСТИ

Системный анализ, информатика и управление

Выпускная квалификационная работа

Научный руководитель:

Доктор физико-математических наук,

Профессор Петросян. О. Л.

Санкт-Петербург

2023

Оглавление

Введение

Актуальность темы диссертации

По сравнению с классической статистикой и математическими методами, методы машинного обучения имеют большие преимущества в крупномасштабных, сложных и нелинейных системах. Поэтому они становятся все более популярными в промышленных приложениях. Системы искусственного интеллекта (ИИ) обычно используются для решения следующих математических задач:

- Классификация: классифицировать входные данные по различным категориям, таким как классификация текста и классификация изображений.
- Регрессия: для прогнозирования значения непрерывных переменных, таких как прогнозирование цены и прогнозирование сетевого трафика.
- Кластеризация: сгруппировать данные на основе их сходства, например, пользовательских предпочтений и классификации данных по функциям.
- Поиск аномалий: для выявления аномальных экземпляров в выборке, например, обнаружение рака и обнаружение аномальных логов.
- Оптимизация: поиск оптимального решения на основе заданной целевой функции, такой как оптимизация сетевого трафика и оптимизация распределения ресурсов.
- Прогнозирование: предсказывать будущие тенденции на основе прошлых данных, таких как прогноз цен на жилье и прогноз потребительского спроса.

Развитие технологий машинного обучения внесло большой вклад в быстрое развитие ИИ. Для машинного обучения, особенно для глубокого обучения, объяснимый ИИ — большой вызов. Глубокие нейронные сети — это «черный ящик»

для всех нас. Алгоритмы ИИ обычно не могут объяснить логику принятия решения. Такие непрозрачные решения недостаточно убедительны, особенно в сферах военной, медицинской и финансовой безопасности, где ставки высоки. Поэтому объяснимый ИИ был бы полезен:

- Для пользователей, когда технология ИИ предназначена для того, чтобы предлагать или помогать принимать решения. Пользователи системы должны иметь возможность понять, почему система предоставляет каждое конкретное решение. Например, врач, ставящий диагноз, должен уметь понимать, почему лечебно-диагностическая система дает такую рекомендацию [1].
- Для разработчиков, чтобы понять «черный ящик» глубокого обучения. Это позволит им улучшать свои методы и модели машинного обучения [2].

Данная диссертация посвящена изучению объяснимых решений в системах искусственного интеллекта большой размерности. Анализируя системы обнаружения аномалий, мы успешно строим объяснимые модели с высокой способностью к обобщению. Мы также изучаем различные методы с разных точек зрения, такие как сэмплирование, кластеризация, иерархические и графические методы, чтобы повысить эффективность объяснимого решения и добиться создания более эффективной объяснимой системы ИИ для многомерных задач.

Обзор результатов в этой области

Применение объяснимого ИИ жизненно важно для современных технологий ИИ. С одной стороны, это делает процесс принятия решений в системах ИИ более прозрачным и понятным, что повышает доверие пользователей к системам и приводит к более широкому внедрению и принятию технологий ИИ, особенно в таких важных областях принятия решений, как здравоохранение и финансы. С другой стороны, разработчики могут исследовать причины ошибок модели, выявлять слабые места в конструкции модели и вносить целевые коррективы в структуру и параметры модели для достижения лучших результатов.

Обнаружение аномалий является одной из важных проблем в области ИИ, которая хорошо изучена в различных областях исследований и приложений. Общей потребностью при анализе наборов данных реального мира является

определение того, какие экземпляры выделяются как непохожие на все остальные. Такие случаи известны как аномалии, и цель обнаружения аномалий (в данной работе мы рассматриваем как аномалии, так и выбросы) состоит в том, чтобы определить все такие случаи на основе данных [3]. Аномалии могут быть вызваны ошибками в данных, но иногда они указывают на новый, ранее неизвестный лежащий в основе процесс. Hawkins в [4] определяет выброс как наблюдение, которое настолько значительно отличается от других наблюдений, что вызывает подозрение, что оно было вызвано другим механизмом. Наиболее распространенными причинами выбросов или аномалий в наборе данных являются *ошибки ввода данных* (человеческие ошибки), *ошибки измерений* (ошибки прибора), *экспериментальные ошибки* (извлечение данных или планирование/выполнение эксперимента), *преднамеренные* (фиктивные выбросы, сделанные для проверки методов обнаружения), *ошибки обработки данных* (манипулирование данными или непреднамеренные изменения набора данных), *ошибки выборки* (извлечение или смешивание данных из ошибочных или несопоставимых источники) и просто наличие новизны в данных. Что касается методов или алгоритмов, обнаружение аномалий следует классифицировать как *обучение с учителем, обучение без учителя, гибридные подходы.* По приложениям обнаружение аномалий можно классифицировать по *обнаружению вторжений, обнаружению мошенничества, обнаружению вредоносных программ, обнаружению медицинских аномалий, обнаружению аномалий в социальных сетях, обнаружению аномалий журналов, обнаружению аномалий больших данных интернета вещей (IoT), обнаружению промышленных аномалий, обнаружение аномалий во временных рядах* и *аномалий при видеонаблюдении.* Более подробную информацию можно найти в недавнем обзоре [5]. Существуют также некоторые потенциальные подходы, используемые для повышения производительности и точности при получении аномалии, такие как успешная модель геометрических преобразований [6] в сочетании с регрессионной моделью [7] и разложением ITO [8] для преодоления временные ограничения.

Алгоритмы обнаружения аномалий часто считаются ограниченными, поскольку они не могут облегчить процесс проверки результатов, выполненных экспертами в предметной области. Это актуальная задача для отрасли. В 2019 г. компания Antwarg использовала фрэймворк SHAP [9] для объяснения обнаружения аномалий. Они рассматривают каждую функцию как игрока и предо-

ставляют пользователям более интуитивное понимание, измеряя вклад каждого игрока в решение. SHAP основан на понятии оптимального значения Шепли [10], которое является хорошо известным понятием из теории кооперативных игр [11] Первоначально значение Шепли определяло, как распределять прибыль, издержки или, в более общем смысле, полезность между игроками, действующими совместно. В случае объяснимого ИИ значение Шепли может показать вклад каждого входного признака в результат системы обнаружения аномалий. Важно отметить, что значение Шепли показывает не только индивидуальный вклад признака в результат система обнаружения, но также показывает вклад признака во все возможные комбинации признаков, который составляет аномалию. Сам подход SHAP для XAI был предложен Лундбергом в [12]. Авторы [13] представляют улучшенный SHAP с использованием метода Baseline Shapley (BShap), который они дополнительно расширяют с помощью интегрированных градиентов в непрерывную область. В статье [14] исследуется зависимость между значениями SHAP путем расширения KernelSHAP для обработки зависимых признаков. В статье [15] авторы описали расширение метода SHAP для деревьев в среде под названием TreeExplainer для изучения структуры глобальной модели с использованием локальных объяснений. Позже в статье [16] описывается метод на основе SHAP для учета прогнозов сигналов временных рядов с использованием сетей долгой краткосрочной памяти (LSTM).

Помимо SHAP, существует несколько других полезных и прикладных алгоритмов для объяснения алгоритмов черного ящика, но в этой диссертации нас особенно интересуют подходы XAI, основанные на использовании значения Шепли:

- LIME — это метод, который интерпретирует прогнозы отдельных моделей на основе построения локальной аппроксимации модели вокруг заданного прогноза [17].
- DeepLIFT (Deep Learning Important FeaTures) [18] – это метод декомпозиции прогноза вывода нейронной сети на конкретный вход путем обратного распространения вкладов всех нейронов в сети в каждый элемент входа.
- LRP (Layer-wise Relevance Propagation) [19] – это метод, придающий объяснительную способность потенциально очень сложным глубоким нейронным сетям. Он действует путем распространения ошибки прогноза в обрат-

ном направлении в нейронной сети с использованием набора специально разработанных алгоритмов распространения.

Более полный и фундаментальный обзор подходов и моделей объяснимого ИИ см. в [20]. Таким образом, объяснимый ИИ можно разделить на следующие типы на основе принципов объяснения:

- Внутренний (объяснимость для моделей): эта ветвь нацелена на использование интерпретируемых моделей для обеспечения объяснимости самого решения, таких как линейная регрессия, дерево решений, байесовская сеть и т.д.

- Объяснение постфактум: эта ветвь предназначена для непосредственного объяснения результатов моделей черного ящика, помогая пользователям понять, почему и как алгоритм приводит к результату Примеры включают LIME, SHAP, DeepLIFT и LRP.

Это общепринятая категоризация объяснимых решений, основанных на технологии, подробно описанной различными авторами [48, 49].

Технология искусственного интеллекта разрабатывалась несколько лет, и в отрасли существует множество приложений, основанных на различных методах. Внутреннего объяснимого метода недостаточно для общего применения для всех из них. Таким образом, многие исследователи предлагали различные методы объяснения постфактум, где решение на основе Шепли является одним из самых популярных методов. Значение Шепли с его свойствами справедливости, модельного агностицизма, локальной и глобальной объяснимости, непротиворечивости и наглядности, эффективно оценивает вклад игроков в области объяснимого ИИ. Несколько статей, в том числе [12, 50], тщательно исследовали эти преимущества. Кроме того, значение Шепли широко применяется в различных областях, таких как прогнозирование [51, 52], обнаружение [53, 54] и классификация [55].

В этой диссертации мы сосредоточимся на изучении интерпретируемых решений с сильными способностями к обобщению на основе значений SHAP, особенно в системе обнаружения аномалий. Однако этот подход сопряжен с проблемами многомерных задач. С одной стороны, многомерные данные означают, что объем вычислений резко возрастает, и корреляция между различными признаками в процессе интерпретации будет препятствовать объяснимому ИИ. С

другой стороны, вычисление значений SHAP является NP-трудной задачей, и по мере увеличения размерности данных, вычислительные затраты растут в геометрической прогрессии.

Вычисление значения Шепли — это NP-сложная задача, требующая рассмотрения $2^N - 1$ комбинаций, где N – это количество признаков. Несколько исследователей начали решать проблему вычисления значения Шепли за полиномиальное время. Гранот предложил древовидную сетевую структуру для полиномиального вычисления значения Шепли [56]. Халкиадакис и его команда сосредоточились на разработке эффективных стратегий для значения Шепли [57], а Кастро и его команда предложили приблизительный метод расчета для него [29]. Позже Кастро и его команда представили метод стратифицированной случайной сэмплировании с оптимальным распределением [58].

В области объяснимого ИИ несколько подходов оценивают значение Шепли для различных алгоритмов: Tree Explainer [15], Deep Explainer (Deeplift + значения Шепли) [12, 59] и Kernel Explainer (Linear Lime + значения Шепли) [12, 17]. Эти объяснимые методы продемонстрировали хорошую эффективность при оценке значения Шепли. Однако большинство из них подходят только для определенных типов алгоритмов. Поэтому мы пытаемся исследовать объяснимые решения на основе значение, которое может быть применено к более широкому кругу алгоритмов.

Подводя итог вышеизложенному, с широким внедрением ИИ в промышленные технологии важно исследовать объяснимые решения ИИ в многомерных системах ИИ для повышения прозрачности и надежности ИИ.

Цели диссертации

Основная цель этой диссертации состоит в том, чтобы изучить и разработать объяснимые решения для многомерных систем искусственного интеллекта. Для достижения этой цели диссертация фокусируется на одной из основных проблем в промышленной области: обнаружение аномалий. В частности, мы изучаем два типа системы обнаружения аномалий, а именно обнаружение аномальных логов и обнаружение рака, и предлагаем новые методы повышения интерпретируемости и эффективности решений многомерного ИИ путем анализа данных и алгоритмов, используемых в этих системах. Поэтому основное внимание в этой диссертации уделяется интерпретируемым решениям самим по себе, а не алгоритмам обнаружения аномалий, с упором как на интерпретируемость, так

и на оптимизацию эффективности.

Диссертация состоит из трех глав. Первая глава направлена на изучение системы обнаружения аномальных логов и использование значения Шепли для объяснения вклада каждой входной функции в выходные результаты. Вторая глава посвящена изучению многомерных систем обнаружения рака и использование значений Шепли для интерпретации влияния функций на результаты обнаружения рака. Мы также разрабатываем и внедряем различные интерпретируемые решения для решения проблемы эффективности интерпретируемого ИИ в многомерных системах. В третьей главе рассматривается, как более быстро и стабильно интерпретировать влияние признаков на результаты обнаружения рака на основе многомерной системы обнаружения рака, изученной в главе 2, с использованием значений Шепли.

Основные задачи

Для достижения поставленных целей обозначим основные задачи данного исследования:

- Изучение и понимание систем ИИ, особенно систем обнаружения аномалий. В этой диссертации мы сосредоточимся на системах обнаружения аномальных логов и обнаружения рака, которые включают три алгоритма: деревья решений, алгоритм DeepLog, и алгоритм Isolation Forest в системе обнаружения рака. Поскольку наши исследования сосредоточены на интерпретируемых решениях в системах ИИ, мы не улучшаем сами алгоритмы обнаружения аномалий, а используем их в качестве инструментов для улучшения наших интерпретируемых решений.

- Изучение и понимание интерпретируемых методов. Чтобы выполнить эту задачу, мы изучили большое количество литературы и получили представление о различных основных интерпретируемых методах ИИ. Поскольку мы стремимся разработать интерпретируемые решения, которые могут применяться ко всем системам ИИ, после изучения различных интерпретируемых методов мы решили сфокусироваться на значениях Шепли.

- Проведение интерпретируемого моделирования и разработка алгоритмических решений на основе соответствующих проблем. В задаче обнаружения аномальных логов в главе 1 мы используем значение Шепли для расчета вклада различных событий в результаты алгоритма Decision Forest. Для

DeepLog мы разработали упрощенное двухуровневое решение в сочетании со значениями Шепли для достижения быстрого расчета путем анализа бизнес-функций. В задаче обнаружения рака, в главе 2, мы систематически модифицируем двухуровневой подход с использованием значений Шепли и используем сэмплирование для значений Шепли для интерпретации влияния входных данных на выходные. В главе 3 мы оптимизируем сэмплирование на основе системы обнаружения рака и разрабатываем алгоритм сэмплирования основанный на графе для дальнейшего повышения эффективности вычислений.

- Разработка показателей сходимости и измерения. Интерпретируемые решения включают сортировку вклада, значение вклада и другие показатели для разумной оценки результатов. Мы пробуем разные методы и предлагаем использовать ранговую корреляцию Спирмена и среднюю абсолютную ошибку (MAE) в качестве показателей сходимости путем анализа важности. Мы также разрабатываем разумные критерии оценки для положительных и отрицательных интервалов вклада, чтобы оценить достоверность и точность результатов.

Научная новизна

В этой диссертации мы исследуем интерпретируемые решения для многомерных задач обнаружения аномалий ИИ.

В главе 1 мы изучаем и анализируем систему обнаружения аномальных логов и предлагаем интерпретируемое решение на основе значения Шепли, которое применяется к деревьям решений и алгоритмам DeepLog на основе нейронных сетей. Чтобы решить проблему больших вычислительных затрат, мы предложили би-уровневый метод, основанный на теории игр в сочетании со значением Шепли, что обеспечивает быстроту вычислений.

В главе 2 мы изучаем и анализируем систему обнаружения рака и предлагаем два интерпретируемых решения, основанных на сэмплировании значений Шепли и улучшенных двухуровневых методах для интерпретации решения на основе Isolation Forest. Оба метода значительно повышают эффективность по сравнению с оригинальным решением.

В главе 3 мы оптимизируем алгоритм сэмплирования и предлагаем улучшенный алгоритм сэмплирования на графе. Мы также дополнительно анализиру-

ем и предлагаем более комплексные показатели сходимости: комбинированную оценку индекса корреляции Спирмена и MAE. Кроме того, мы предлагаем более подходящий способ оценки точности, основанный на интервалах сортировки или измерениях положительных/отрицательных вкладов.

В целом, научная новизна данного исследования заключается в разработке различных решений, ориентированных на скорость сходимости и вычислительную эффективность, для задач обнаружения аномалий ИИ высокой размерности.

Методы исследования

В этой диссертации мы используем различные области исследования, в том числе:

- Теория игр (значение Шепли и двухуровневый подход).
- Теория графов (задача о кратчайшем пути).
- Машинное обучение (алгоритм Isolation Forest, алгоритм дерева решений, алгоритм кластеризации с ограничением k-means).
- Нейронные сети (алгоритм DeepLog).
- Статистика (коэффициент корреляции Пирсона, индекс корреляции Спирмена, метрика MAE).
- Теория вероятностей (случайное блуждание).
- Информатика (программирование на Python).

Теоретическая и практическая значимость

Технология искусственного интеллекта сегодня является одной из самых актуальных тем в промышленных приложениях, и она широко применяется в различных областях, таких как здравоохранение, военные, управление, финансы и производство. Технология искусственного интеллекта в основном используется для решения задач оптимизации, теории графов, статистики, теории вероятности и т. д. По сравнению с традиционными математическими методами, технология ИИ обладает более широкими возможностями в решении многомерных и сложных задач, при этом существенно зависит от данных, а сами алгоритмы ИИ не обладают прозрачностью, что подчеркивает важность интерпретируемых решений.

Ключевым моментом исследования этой диссертации являются интерпретируемые решения для многомерных задач ИИ, которые имеют широкий спектр областей применения, а также являются одной из самых актуальных тем исследований в промышленных приложениях. Диссертация в основном посвящена обнаружению аномалий в области ИИ, что является критической проблемой в таких областях, как здравоохранение, финансы, вооруженные силы и автономное вождение. Однако реальные промышленные проблемы часто связаны с огромными масштабами, что приводит к проблеме проклятия размерности. В этой работе предлагаются гибридные методы решения, такие как сэмплирование и двухуровневые методы с точки зрения моделирования бизнес-задач и разработки алгоритмов. Эти методы имеют значительные преимущества в производительности при решении многомерных задач и учитывают способность решений к обобщению, что делает их подходящими для различных задач ИИ. Они также имеют хорошую теоретическую основу и прикладную ценность.

Краткое описание структуры диссертации

Эта диссертация состоит из введения, трех основных глав, заключения и справочных разделов. Введение дает обзор темы исследования, включая базовые знания, текущий статус исследования и связанную с ним техническую основу, цели исследования, основные задачи, научную новизну, исследования методы, теоретическое и практическое значение, структура диссертации, информация о публикациях и благодарности.

В главе 1 диссертации основное внимание уделяется исследованию на основе системы регистрации аномалий логов, в разделе 1.1 представлены проблемы системы обнаружения аномалий и двух алгоритмов, необходимых для обнаружения аномалий: дерева решений и алгоритма DeepLog на основе нейронных сетей. В разделе 1.2 мы разрабатываем подходы для интерпретации моделей, используя значение Шепли для задачи, и представляем двухуровневое решение в сочетании со значением Шепли для алгоритма DeepLog. В разделе 1.3 анализируются результаты объяснения обоих решений, полученных моделями. В разделе 1.4 обобщаются результаты исследования и выводы главы 1.

В главе 2 диссертации основное внимание уделяется исследованию, основанному на системе обнаружения рака. В разделе 2.1 описывается проблема обнаружения рака и предложенный для обнаружения алгоритм Isolation Forest. В разделе 2.2 предлагается, как построить интерпретируемые модели и алго-

ритмы для задач обнаружения рака с использованием расчета значений Шепли на основе сэмплирования и двухуровневого подхода в сочетании со значением Шепли для интерпретируемых решений. В разделе 2.3 представлены данные о раке из открытых источников, сравниваются результаты тестирования как двухуровневого метода, так и сэмплирования, а также глобальная и локальная интерпретируемость. Раздел 2.4 обобщает результаты исследования и выводы главы 2.

В главе 3 продолжается изучение системы обнаружения рака. В разделе 3.1 подчеркивается необходимость дальнейших исследований для улучшения метода сэмплирование. В разделе 3.2 предлагается алгоритм сэмплирования, основанный на карте взаимосвязей. Предлагается использование коэффициента корреляции Пирсона для построения матрицы взаимосвязей, а для сэмплирования используется смещенное случайное блуждание. Индекс корреляции Спирмена и MAE предлагаются для обеспечения стабильности и точности результатов. Раздел 3.3 описывает данные и выполненные тесты. Раздел 3.4 суммирует результаты исследования и выводы главы 3.

В заключительной главе обобщаются результаты исследования и выводы, а также некоторые мысли о будущей работе.

Последним разделом дипломной работы является справочный раздел, в котором указана вся изученная литература и ссылки на нее.

Результаты, представленые на защиту

- Разработка и реализация интерпретируемых подходов с использованием значения Шепли в системах обнаружения аномальных логов.
- Разработка и реализация интерпретируемых методов с использованием двухуровневого подхода и значения Шепли на основе эмпирических знаний.
- Разработка и применение интерпретируемых алгоритмов с использованием сэмплирования значения Шепли в задаче обнаружении рака.
- Проектирование и реализация иерархического алгоритма k-means для кластеризации.
- Разработка и применение иерархических двухуровневых подходов на основе бизнес-знаний в задаче обнаружения рака.

- Разработка и построение матрицы отношений для интерпретируемых игровых моделей с использованием коэффициента корреляции Пирсона.
- Разработка и реализация сэмплирования на основе смещенного случайного блуждания на графе.
- Разработка и реализация метрики сходимости сэмплирования с использованием индекса корреляции Спирмена и MAE.
- Разработка и реализация интерпретируемых алгоритмов с использованием сэмплирования на графе.
- Разработка и реализация методов оценки интерпретируемости результатов на основе положительных и отрицательных вкладов.

Верификация и опубликованные результаты

Исследование в этой диссертации в основном было проведено и опубликовано первым автором, а последняя из статей была представлена на конференцию и в настоящее время находится на рассмотрении и доработке.

1. РИНЦ: Zou Jinying, Xu Feiran, Petrosian Ovanes. Explainable AI: Using Shapley Value to Explain the Anomaly Detection System Based on Machine Learning Approaches.ПРОЦЕССЫ УПРАВЛЕНИЯ И УСТОЙЧИВОСТЬ, 2020, 355-360.
2. SCOPUS Q4: Jinying Zou, Ovanes petrosian. Explainable AI: Using Shapley Value to Explain Complex Anomaly Detection ML-Based Systems //Machine Learning and Artificial Intelligence: Proceedings of MLIS 2020. – 2020. – T. 332. – C. 152.
3. SCOPUS Q1: Zou, J., Xu, F., Zhang, Y., Petrosian, O. Krinkin, K. High-Dimensional Explainable AI for Cancer Detection. 1 Sep 2021, In: International Journal of Artificial Intelligence. 19, 2, p. 195-217 23 p.
4. Scopus: Explainable AI: Graph Based Sampling Approach for High Dimensional AI System. Jinying Zou, Feiran Xu, Yin Li, Ovanes Petrosian.(It is sent and had passed the review round. Will published in Springer series Lecture Notes in Networks and Systems.)

Благодарности

Я хотел бы выразить благодарность моему научному руководителю Петросяну Ованесу Леоновичу за его неоценимую помощь и поддержку на протяжении всех четырех лет моей учебы и работы. В процессе исследования мы столкнулись со многими проблемами, и многие идеи казались неосуществимыми как на этапе проектирования, так и на этапе реализации. Тем не менее, мой руководитель оказал большую поддержку и руководство в плане проектирования решений, разработки алгоритмов и поддержания позитивного настроя. Кроме того, из-за конфликтов между работой и учебой я пережил период нестабильности и негатива. Я хотел бы поблагодарить мою жену и наших трех кошек за их эмоциональную поддержку и ободрение.

Наконец, я хотел бы поблагодарить всех авторов и исследователей, упомянутых в этой диссертации, а также организации и проектные группы, поддержавшие эту работу:

- Работа, описанная в главе 1, выполнена при поддержке Российского фонда фундаментальных исследований (РФФИ) в соответствии с проектом № 18-00-00727 (18-00-00725).

- Работа в главе 2 выполнена при поддержке Министерства науки и высшего образования Российской Федерации Договором № 075-15-2020-933 от 13.11.2020 о предоставлении гранта в виде субсидии от федерального бюджета на реализацию государственной поддержки создания и развития научного центра мирового уровня «Павловский центр» «Интегративная физиология для медицины, высокотехнологичное здравоохранение и стрессоустойчивые технологии».

- Работа в главе 3 выполнена при поддержке Санкт-Петербургского государственного университета, ID проекта: 94062114

Глава 1

Объяснимый метод интерпретации решений ИИ-системы обнаружения аномальных логов

В этой главе мы применяем подход Шепли к двум системам обнаружения аномалий с разными структурами. Во-первых, это дерево решений [21], [22], в котором мы рассматриваем один признак как игрока (подход игрок-признак). Наше отличие от подхода Антварга [9] в том, что мы рассматриваем различные события в системе обнаружения аномалий как игрока на основе самих данных без учета модели алгоритма. Мы также не используем структуру SHAP [9], а разрабатываем собственную структуру на основе значения Шепли. Вторая система обнаружения аномалий, которую мы хотим объяснить, — это DeepLog [23]. Текущее основное исследование объяснимого ИИ, связанного с Шепли, рассматривает один признак как игрока, а затем анализирует вклад каждого игрока в результат. Но этот подход имеет ограничения для класса систем обнаружения аномалий, где аномалия может быть результатом не только одного признака, но и последовательности признаков. Поэтому мы рассматриваем последовательность событий в качестве игрока для моделирования. Ключевой проблемой является большое количество [24] коалиций в подходе игрок-последовательность по сравнению с подходом игрок-признак. Чтобы избежать проблемы большого количества игроков-последовательностей, мы используем двухуровневый метод для вычисления значения Шепли. Родственный метод кооперативных коалиционных игр можно найти в [25]. На первом уровне в качестве игрока рассматривается одно событие, на втором уровне в качестве игрока рассматривается последовательность из двух событий.

В разделе 1.1 мы кратко представляем алгоритмы как для дерева решений, так и для DeepLog. В разделе 1.2 мы описываем значение Шепли и то, как его применять для объяснения дерева решений и DeepLog. В разделе 1.3 мы прикрепляем результаты нашего моделирования. В разделе 1.4 представлены выводы и обсуждения для будущей работы.

Все наши исследования основаны на проекте с открытым исходным кодом «Loglizer», который представляет собой набор инструментов для анализа журналов на основе машинного обучения для автоматического обнаружения аномалий [26].

1.1 Система обнаружения аномального ИИ

1.1.1 Решение на основе машинного обучения: дерево решений

В 2004 году Майк Чен и другие предложили использовать дерево решений для классификации неудачных и успешных запросов для поиска аномалий в системных логах [21]. Дерево решений представляет собой древовидную структуру, состоящую из узлов и ветвей. Узлы делятся на листовые узлы (представляющие определенную категорию) и внутренние узлы (представляющие определенный атрибут или функцию). Ветви представляют собой тестовый вывод. Основная идея дерева решений заключается в использовании энтропии в качестве меры для выбора атрибутов его узлов. Каждый узел выбирает атрибуты с наибольшим увеличением, то есть атрибут с наименьшим значением энтропии. Когда энтропия равна нулю, все экземпляры в узле считаются одним и тем же кластером. Ниже представлена краткая версия алгоритма дерева решений обнаружения аномалий, примененного к анализу системных логов:

1. Выбрать событие в качестве корня. Лучший корень будет выбран в соответствии с полученной информацией. Уравнение для расчета энтропии прироста информации показано в уравнении 1.1 как показано ниже:

$$G(D,a) = H(D) - H(D \mid a) = H(D) + \sum_{i=1}^{N} \frac{|D^i|}{|D|} H(D^i), \qquad (1.1)$$

где $G(D,a)$ — прирост информации, $H(D)$ — сумма информационной энтропии для всех признаков множества D, $H(D \mid a)$ — условная информационная энтропия при условии признака a.

2. Разделить выборки на два поддерева и найти максимальное усиление.

3. Выполнять итерации (от шага 1 до шага 3) до тех пор, пока во входном наборе данных не останется ни событий, ни признаков.

4. Теперь каждая ветвь дерева представляет собой результат прогнозирования, показывающий, является ли лог аномальным или нормальным.

1.1.2 Решение на основе нейронной сети: DeepLog

DeepLog — это управляемый данными алгоритм, который использует большое количество системных логов для обнаружения аномалий. Основная интуиция, стоящая за дизайном DeepLog, исходит из обработки естественного языка: рассматривать записи лога как элементы последовательности, которые следуют определенным шаблонам и грамматическим правилам [23]. В отличие от метода счетчика сообщений лога, DeepLog представляет собой глубокую нейронную сеть, которая использует долговременную и кратковременную память для моделирования последовательностей лога. Следовательно, важность последовательности событий больше, чем количество сообщений о событиях. Таким образом, в этом исследовании последовательность рассматривается как игрок, объясняющий каждое конкретное решение. Таким образом, мы можем проанализировать, какие последовательности вносят больший вклад в точность предсказания и являются ли последовательности значимыми.

На рисунке 1.1 показана архитектура DeepLog, состоящая из двух частей: обучения и обнаружения. Данные обучения для DeepLog поступают из логов системы. Лог объединяется из ключа лога и вектора значений параметров. На этапе обучения лог нужно сначала разобрать. Затем полученную последовательность событий можно использовать в качестве входных данных для обучения модели обнаружения. После завершения обучения система может судить о том, является ли лог нормальным в соответствии с ключом. Если это нормально, DeepLog дополнительно проверит вектор значений параметра. Если вектор значений параметра является аномальным, он будет помечен как аномальный лог.

DeepLog использует долговременную кратковременную память (LSTM) в рекуррентной нейронной сети (RNN) в качестве основы для наблюдения за долговременной зависимостью последовательности, рисунок 1.3. Верхняя часть пред-

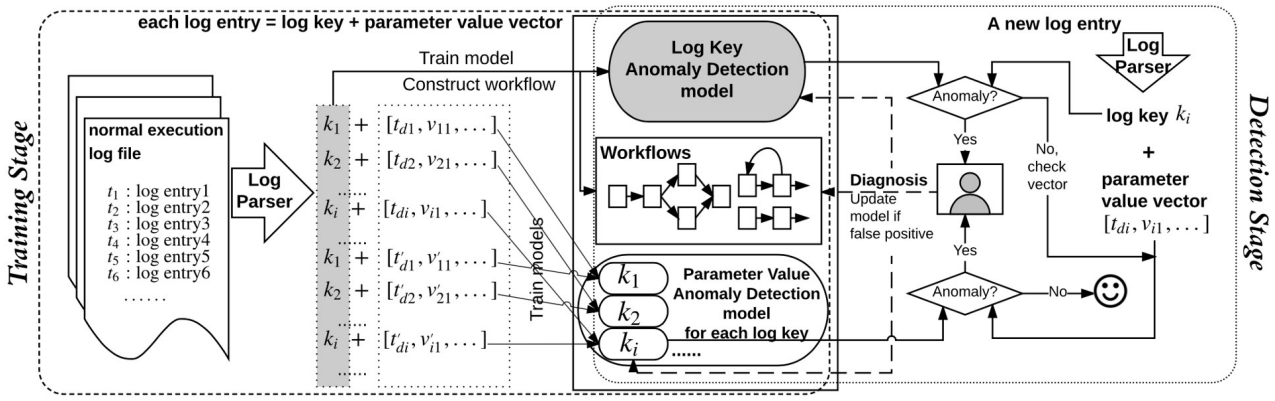

Рис. 1.1: Архитектура DeepLog

ставляет собой блок LSTM с одной природой. Каждый блок LSTM запоминает состояние своего входа как вектор фиксированной размерности. m_{t-i} является примером ввода. В центре показана серия блоков LSTM, где в каждой ячейке поддерживаются скрытый вектор H_{t-i} и вектор состояния C_{t-i}. И скрытый вектор, и вектор состояния будут переданы следующему в качестве начального ввода для сохранения исторической информации. В нижней части рисунка 2 показан пример DeepLog с двумя скрытыми слоями схемы глубокой нейронной сети. Входной слой кодирует n возможных лог-ключей из K в виде однократных векторов. Выходной слой, использующий стандартную полиномиальную логистическую функцию, переводит n слой скрытого состояния в функцию распределения вероятностей [23].

1.2 Объяснимая модель и алгоритм

1.2.1 Ценность Шепли и объяснимая модель

В классической теории кооперативных игр предполагается, что игроки сотрудничают и вместе получают общее вознаграждение, а затем распределяют это вознаграждение между собой [27]. Чтобы разделить общее вознаграждение, вводится понятие импутации. В этой главе в качестве импутации мы используем классическое кооперативное решение значения Шепли [10]. Явное уравнение для значения Шепли представлено в уравнении 1.2 как показано ниже:

$$\varphi_i = \sum_{S|i \in S \subseteq N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[v(S) - v(S \setminus \{i\})] \tag{1.2}$$

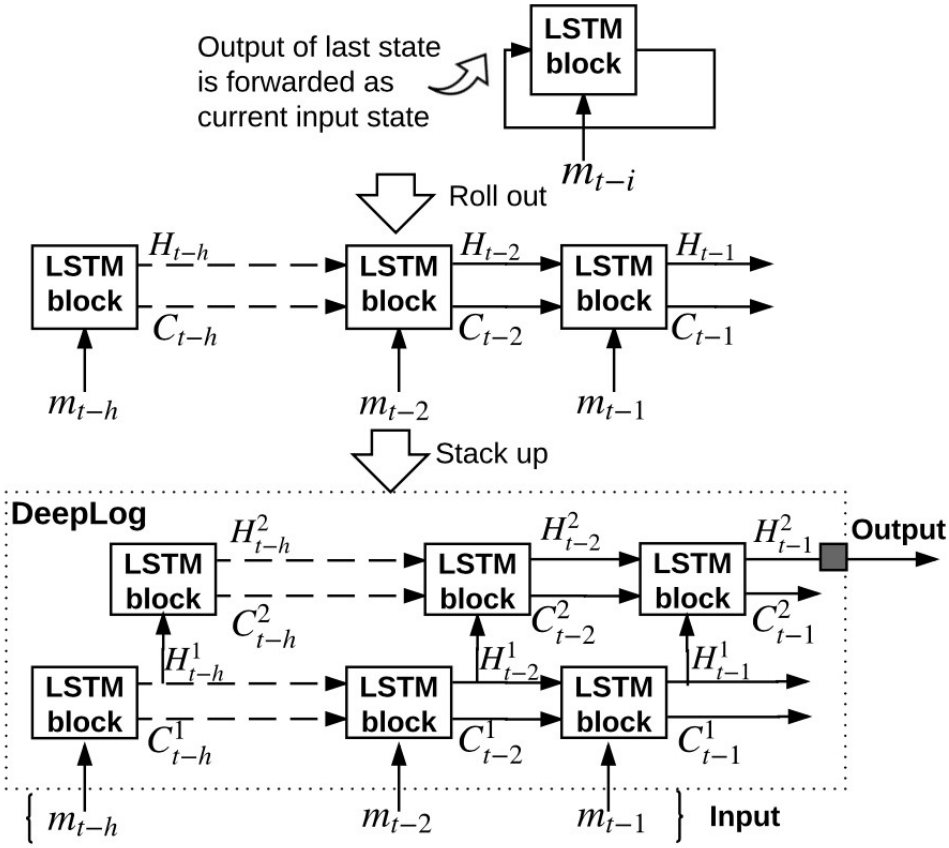

Рис. 1.2: Представление об обнаружении аномалий с использованием составного LSTM.

где i — игрок, N — множество всех игроков в кооперативной игре, $S \subseteq N$ — коалиция игроков, $|S|$ — количество игроков в коалиции S, $V(S)$ — характеристическая функция коалиции $S \subseteq N$, задающая общий выигрыш коалиции S.

Чтобы использовать значение Шепли в любой области, необходимо вычислить значения характеристической функции $V(S)$ для каждой коалиции $S \subseteq N$. Более подробная информация о том, как его рассчитать для объяснения обнаружения аномалий в лог-системе, будет представлена в следующих разделах. В теории кооперативных игр, как видно из уравнения (1.2), значение Шепли игрока i показывает взвешенную сумму вкладов игрока i в вознаграждение за сотрудничество каждой коалиции S из N (член $[v(S) - v(S \setminus \{i\})]$). Левый множитель в произведении (1.2) определяет вероятность образования самой коалиции S, поэтому чем меньше вероятность, тем менее важен индивидуальный вклад игрока i в сотрудничество.

Используя значение Шепли, общее вознаграждение распределяется между игроками ($\sum_{i=1}^{n} \varphi_i = V(N)$). Вообще говоря, если вклад игрока в сотрудниче-

ство велик, то и значение его вменения велико. В машинном обучении подход значений Шепли может применяться для объяснения вклада каждого значения функции в общее решение.

1.2.2 Объяснимое решение для дерева решений

Основная идея дерева решений заключается в использовании нисходящего рекурсивного метода для использования информационной энтропии в качестве меры для построения наиболее быстро падающего значения энтропии. Значение энтропии в листовом узле равно нулю. В это время экземпляры в каждом конечном узле принадлежат к одному и тому же классу. Другими словами, сущностью дерева решений является набор правил «если-то». Дерево решений делит пространство признаков на непересекающиеся единицы или регионы.

Для того чтобы применить подход, основанный на значении Шепли, на первом шаге необходимо вычислить значения характеристической функции для каждой коалиции $S \subseteq N$, где N и S — множество и подмножество всех особенности или уникальные события в системном логе соответственно. Смысл характеристической функции $V(S)$ коалиции S для системы обнаружения аномалий состоит в значении обнаружения аномалии или вероятности аномалии только по событиям из множества S. После вычисления характеристической функции для каждой коалиции S можно вычислить значение Шепли и, в результате, объяснить результат дерева решений, объяснив вклад каждого признака. Алгоритм вычисления значения Шепли для обнаружения аномалий дерева решений:

1. Определить проигрыватель признаков $n = |N|$ из набора данных D.
2. Выбрать коалицию $S \subseteq N$ игроков. Общее количество рассматриваемых коалиций равно $\sum_{k=0}^{n} C_n^k$.
3. Запустить алгоритм дерева решений для каждой коалиции $S \subseteq N$, чтобы получить значение характеристической функции $V(S)$ или точность обнаружения аномалии.
4. Повторять шаги $2 - 3$ до тех пор, пока не будут вычислены все значения характеристической функции $V(S)$.
5. Используя уравнение (1.2), рассчитать значение Шепли для всех игроков.

1.2.3 Объяснимое решение для DeepLog

Подход DeepLog позволяет пользователю не только найти признак или событие, являющееся аномалией, но и определить последовательность событий, которые могут привести к аномалии в системном логе. Следовательно, объяснимый ИИ должен решить и эту проблему, введя объяснение не только для набора отдельных событий, но и для последовательности событий.

Здесь мы также определяем точность предсказания аномалий для набора признаков S с помощью характеристической функции значений $V(S)$. Но мы рассматриваем события 2 и целевое событие 1 как признак или игрока (рисунок 1.3). Все последовательности будут разделены и отсортированы по целевому событию. Мы вычислим значение характеристической функции $V(S)$ для наборов S событий последовательности в системном логе вместо набора одиночных событий, чтобы проверить вклад каждой последовательности. Рабочий процесс DeepLog представлен ниже:

1. Анализ лога. Целью анализа лога является преобразование неструктурированных сообщений лога в структурированное отображение событий, на основе которого можно применять сложные модели машинного обучения.

2. Извлечение признаков: обычно структурированные логи можно разрезать на короткие последовательности логов через интервальные окна, скользящие окна или окна диалога. В DeepLog мы используем скользящие окна для извлечения признаков и векторизируем каждую извлеченную последовательность.

3. Обнаружение аномалий: после того, как модель получена с помощью обучающих данных, на извлеченных данных выполняется прогнозирование аномалий.

Согласно рисунку 1.3, короткие последовательности событий будут извлечены из необработанных длинных последовательностей системы, чтобы составить набор признаков или игроков. Эта процедура отличается от процедуры в алгоритме дерева решений. В алгоритме DeeplLog рассматривается подход последовательности для измерения вклада каждой последовательности в обнаружение аномалии. Последовательный подход приводит к проблеме большого количества коалиций или проклятию размерности. Чтобы избежать этой проблемы,

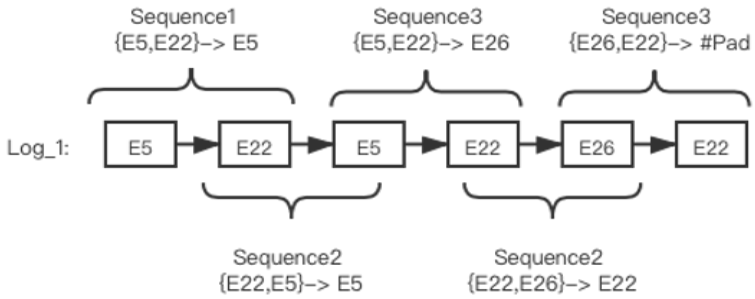

Рис. 1.3: Пример последовательного подхода.

мы используем двухуровневый подход для вычисления значения Шепли или оценки последовательности для обнаружения аномалий. На первом уровне в качестве признака или игрока мы рассматриваем только целевое событие, а на втором уровне для каждого фиксированного целевого события мы рассматриваем в качестве игрока последовательность двух событий. Поэтому количество игроков или признаков на первом уровне соответствует количеству уникальных целевых событий $E \in N$ в системном логе и равно $|N|$. На втором уровне количество признаков или игроков различно в зависимости от связанного с ним целевого события E и равно количеству уникальных последовательностей двух событий $\{E_i, E_j\} \in N_E$ перед целевым событием E ($|N_E|$), $E \in N$. Далее для первого уровня вычисляются значения характеристической функции $V(S)$, $S \subseteq N$ с использованием того же подхода, что и для алгоритма дерева решений. А для второго уровня значения характеристической функции $V_E(S)$, $S_E \subseteq N_E$ вычисляются для каждого фиксированного целевого события $E \in N$. Более поздние значения Шепли рассчитываются с использованием уравнения 1.2 как для первого уровня φ^t, так и для второго уровня с последовательностями событий φ_s. Результирующее значение Шепли для каждой последовательности событий получается путем умножения значений Шепли на первом и втором уровнях. Алгоритм расчета значения Шепли для обнаружения аномалий DeepLog:

1. Используя скользящее окно = 3, извлечь признаки (целевые события и соответствующие два события последовательности).

2. Определить G_n как группу последовательных игроков, где n - количество уникальных целевых игроков.

3. Выбрать i-ю коалицию, i от 0 до $\sum_{k=0}^{n} C_n^k$.

4. Запустить алгоритм DeeplLog, чтобы получить точность предсказания.

5. Обновить характеристическую функцию для i-й коалиции с предсказанием точности. i++, повторяйте шаги с 3 по 5, пока не будут получены все характеристические функции.

6. Используя уравнение (1.2), вычислить значение Шепли для всех игроков.

1.3 Результаты симуляции

В этом разделе мы представляем результаты моделирования, объясняющие решение предварительно обученного дерева решений и DeepLog.

1.3.1 Объяснимые результаты дерева решений

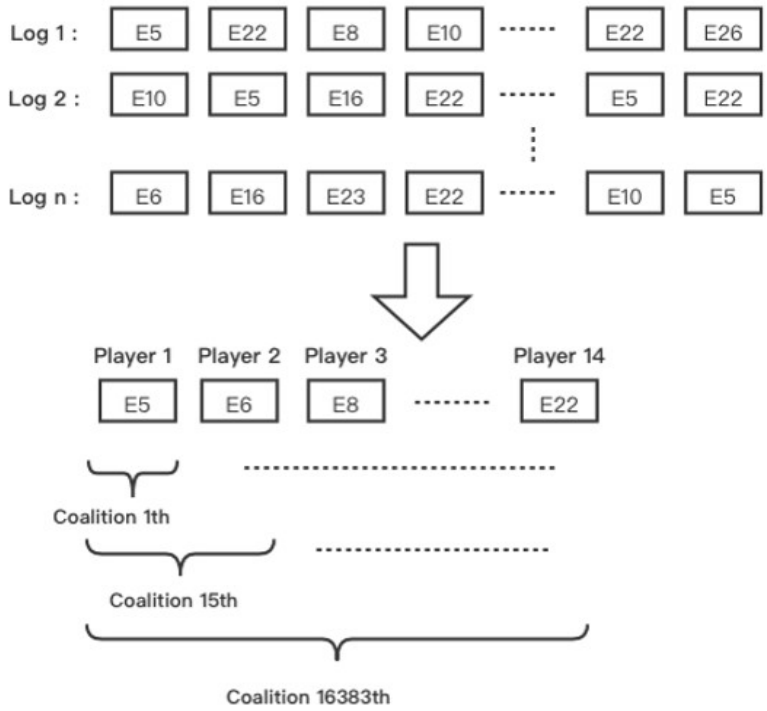

Рис. 1.4: Связь системного лога с компонентами или совместной игрой.

Для дерева решений мы рассматриваем два набора входных данных. Оба системных лога состоят из $n = 3900$ экземпляров системного лога с 20 уникальными событиями. Начальный системный лог фильтруется, чтобы сократить количество уникальных событий до 14, поскольку остальные 6 появляются редко. Следовательно, общее количество коалиций можно вычислить следующим образом: $\sum_{k=0}^{14} C_{14}^{k} = 16383$. Процесс определения игроков и коалиций с помощью

исходного системного лога представлен на рисунке 1.4. Используя уравнение 1.2 и значений характеристической функции $V(S)$ вычисляем значение Шепли. Ниже на рисунках 1.5 и 1.6 показаны результаты для двух тестовых наборов данных. Здесь определяется вклад каждого события системного лога в решение дерева решений с использованием тестовых наборов данных.

На рисунке 1.5 показано, что событие $E11$ играет наиболее важную роль в обнаружении аномалии для набора данных 1 для дерева решений. Важным выводом для набора входных данных 1 является то, что даже если мы удалим из системного журнала все события, кроме события $E11$, обнаружение аномалии все равно будет хорошим, поскольку значение Шепли для события $E11$ равно $0,9636$. На рисунке 6 обратите внимание на результаты моделирования для набора входных данных 2. Здесь важную роль играют события $E9$, $E11$ и $E26$. Это означает, что каждое из этих событий вызвало аномалию в системном логе, значения Шепли для событий $E9$, $E11$ и $E26$ равны $0,322$, $0,313$, $0,317$ соответственно.

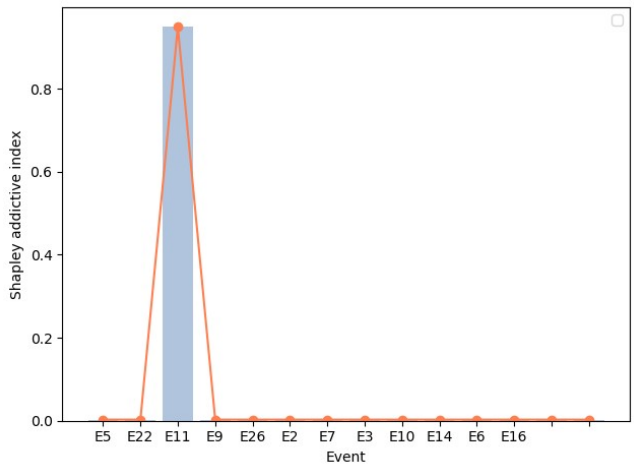

Рис. 1.5: Значение Шепли с набором входных данных 1.

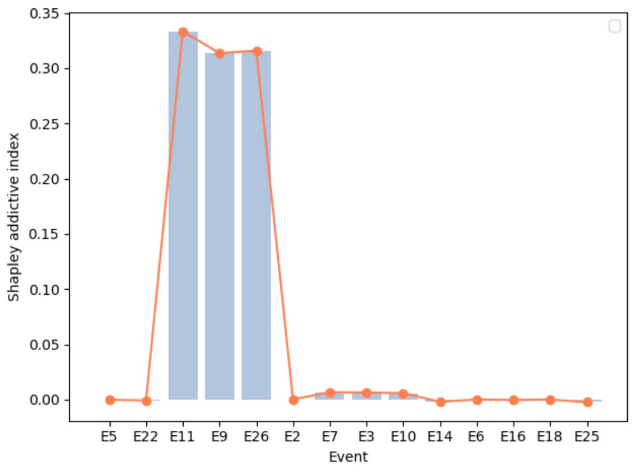

Рис. 1.6: Значение Шепли с набором входных данных 2.

1.3.2 Объяснимые результаты DeepLog

На рисунке 1.3 показан процесс двухуровневого метода. Длина окна для второго уровня равна 3.

После извлечения признаков мы получаем новую структуру данных. Вероятность события после последовательности является ключом к обнаружению аномалии. Низкая вероятность означает аномалию, а высокая вероятность означает

нормальную последовательность. На рисунке 1.7 также есть элемент «#Pad», означающий конец последовательности. Мы можем контролировать, участвуют ли игроки в игре локального уровня, установив событие как Nan. На втором уровне все последовательности с уникальным целевым событием будут считаться игроками.

Набор тестовых данных содержит 1875 экземпляров, которые содержат 15000 последовательностей событий. После очистки данных получаем 7 уникальных событий. Для реализации процедуры, описанной в разделе 1.3, мы рассматриваем две кооперативные игровые модели: модель игры первого уровня для групп событий, модель игры второго уровня для последовательностей событий в каждой группе.

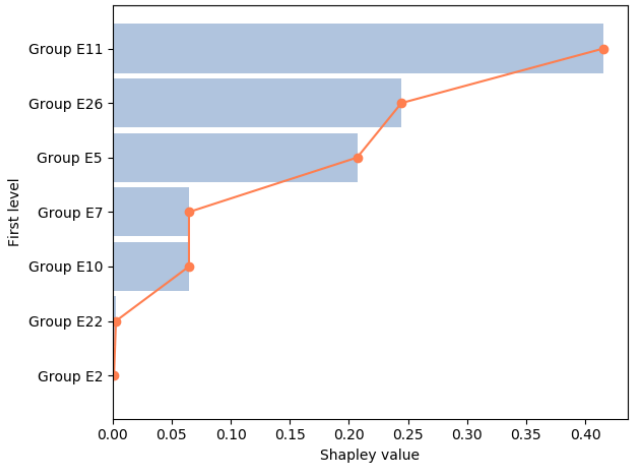

Рис. 1.7: Значение Шепли для групповых игроков в игре первого уровня.

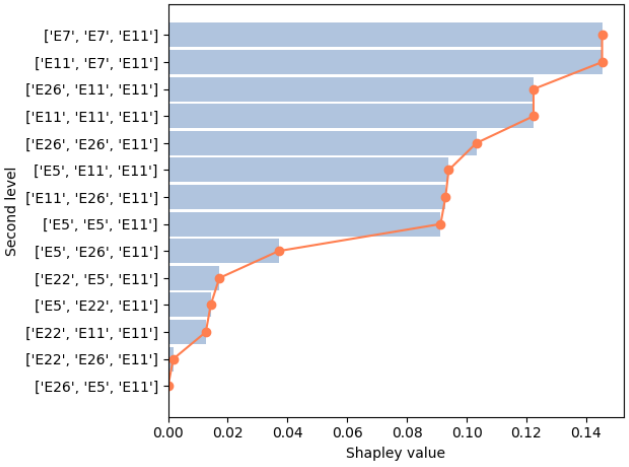

Рис. 1.8: Значение Шепли для последовательных игроков в игре второго уровня для целевого события $E11$.

В игровой модели первого уровня мы определяем игрока как группу последовательностей событий, которые соответствуют определенному целевому событию из исходного системного лога, см. рисунок 3. Затем вычисляем значения характеристической функции и значения Шепли. Важно отметить, что это еще не объясняющий результат для решения DeepLog, соответствующего набору входных данных. На рисунке 1.7 показан вклад каждой группы последовательностей, соответствующих целевым событиям: $E11$, $E26$, $E5$, $E7$, $E10$, $E22$, $E2$. Легко видеть, что вклад $0,415$ в результат DeepLog группы $E11$ самый большой, а вклад группы $E2$ всего $0,0011$.

Игровая модель второго уровня строится отдельно для каждой группы, относящейся к целевому событию. Игрок в игре второго уровня представляет собой

последовательность событий с фиксированным целевым событием. Используя значение Шепли, можно определить вклад каждой последовательности событий в обнаружение аномалий группы событий. На рисунке 1.8 показан вклад каждой последовательности событий в обнаружение аномалии для группы событий, связанных с целевым событием $E11$. Аналогичные результаты получаются для целевых событий $E2$, $E5$, $E7$, $E10$, $E22$, $E26$.

Наконец, когда значения Шепли как для игровых моделей первого, так и для второго уровня (рисунки 1.7 и 1.8) получены, значение Шепли для каждой последовательности событий вычисляется с использованием подхода, описанного в разделе 1.3. Смысл этого значения Шепли — вклад последовательности событий в обнаружение аномалий DeepLog с использованием набора тестовых данных. На рисунке 1.9 показан вклад каждой последовательности событий в обнаружение аномалий DeepLog с использованием набора тестовых данных. Легко видеть, что последовательности $\{E7, E7, E11\}, \ldots, \{E5, E5, E11\}$ вносят наибольший вклад в обнаружение аномалий. Это означает, что в этих последовательностях сосредоточена аномалия системного лога из набора тестовых данных. Пользователь системы DeepLog должен детально проверить, какова природа этих последовательностей, чтобы сделать вывод.

1.4 Выводы к главе 1

В этой главе мы применяем подход значений Шепли к статическому моделированию машинного обучения, в частности к дереву решений и алгоритмам DeepLog. Как показано в разделе моделирования, наш подход может объяснить конкретное решение, используя как отдельные функции, так и последовательности событий. Как показано на рисунке 1.9, пользователь системы обнаружения аномалий, имеющей объяснительный модуль, может легко найти причину аномалии (конкретное событие или последовательность событий), а не только получить сигнал тревоги об аномалии.

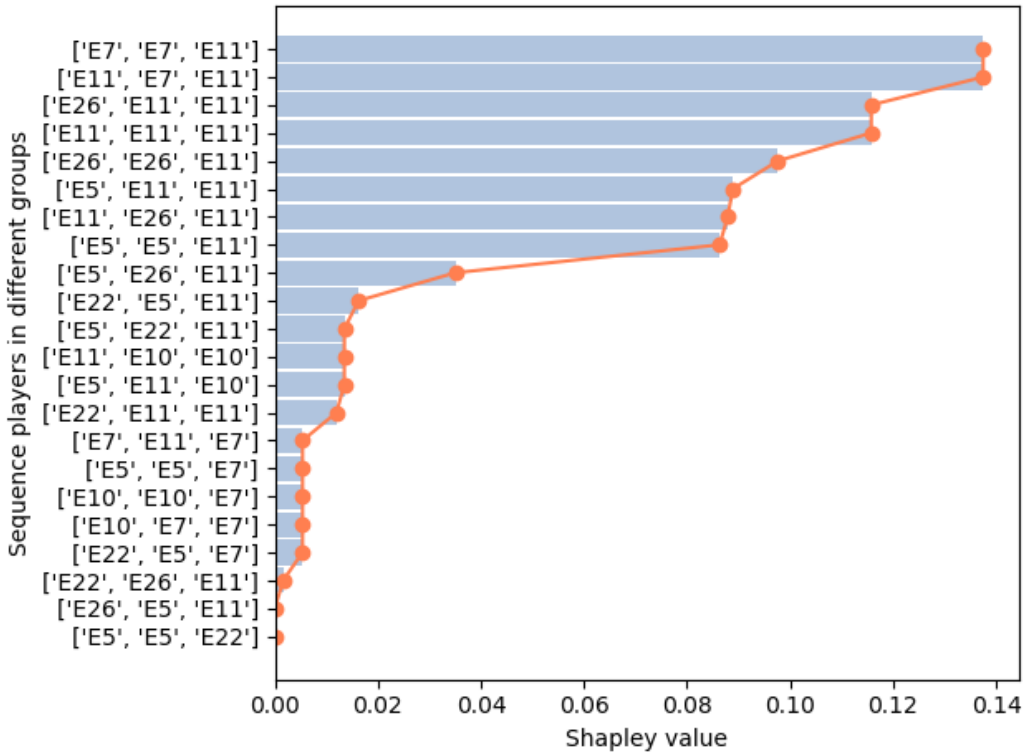

Рис. 1.9: Значение Шепли для результата обнаружения аномалии с помощью DeepLog с набором тестовых данных.

Глава 2

Объяснимый искусственный интеллект: системы большой размерности для обнаружения рака

В настоящее время возможности использования искусственного интеллекта в распознавании и прогнозировании достигли очень высокого уровня. Несколько исследователей уже предложили некоторые подходы ИИ для обнаружения и лечения рака (Kuswanto H et al., 2019) [39]. Однако нельзя игнорировать проблемы этики (Cath C et al., 2018) [35], доверия (Lui A et al., 2018) [42] и предрассудков (Challen R et al., 2019) [36], неизбежно возникающие при применении ИИ в судебной практике, медицинском обслуживании и деятельности, связанной с этническими меньшинствами. Таким образом, объяснимый ИИ необходим для получения достоверных результатов диагностики рака, основанных на обнаружении аномалий. Классификация методов оъяснимого ИИ была подробно рассмотрена в предыдущих обзорах (Adadi A et al., 2018) [31], где эти методы укрупненно делились на два типа: глобальное объяснение и локальное объяснение.

Как правило, обнаружение аномалий — это метод, используемый для выявления аномальных паттернов (выбросов), которые не соответствуют ожидаемому поведению (Patcha A et al., 2007) [43]. По мнению Хокинса, так как выбросы сильно отличаются от других наблюдений в выборке, им нельзя “доверять” (Hawkins DM, 1980) [4]. В случае анализа медико-физиологического данных это означает, что если значения физиологического индекса пациента будут отклоняться от наблюдений других пациентов, то его вероятность заболевания раком значительно возрастет.

В этой главе мы предлагаем два решения. Первое – сэмплирование. Второе – двухуровневый подход, который является нашим вкладом в проблему высокого измерения. Оба метода реализованы для обнаружения аномалий рака. Двухуровневый подход и сэмплирование сравниваются друг с другом, см. заключение ниже:

- Во-первых, двухуровневый подход обладает лучшей масштабируемостью. В игре с N игроков всего $N^2 - 1$ перестановок. Чем больше игроков, тем больше выборок требуется в сэмплировании, чтобы поддерживать требуемую точность. В то же время результат двухуровневого подхода зависит только от размера групп.

- Во-вторых, двухуровневый подход может сочетаться с другими подходами объяснимого ИИ, такими как SHAP, LIME и т.д., для решения более сложных задач.

Двухуровневый подход и сэмплирование используются для локального и глобального объяснения Isolation Forest. Локальное объяснение используется для объяснения результатов обнаружения аномалии для одного конкретного пациента. Это объяснение может повысить доверие врачей и пациентов к системе прогнозирования рака и объяснить причину прогнозирования аномалии. Локальное объяснение может предоставить индивидуальный отчет о прогнозе рака для каждого пациента, что крайне важно для постановки диагноза врачом. Глобальное объяснение — это объяснение системы обнаружения аномалий в целом. Обычно оно может быть выполнено с использованием всей выборки, которая использовалась для обучения системы обнаружения аномалий. Инженеры по машинному обучению могут полагаться на глобальные объяснения, чтобы найти потенциальные проблемы и повысить производительность систем машинного обучения.

Ожидается, что в этой главе будут достигнуты следующие цели:

- Разработать методы, которые могут разумно объяснить многомерную систему ИИ.

- Сравнить производительность двухуровневого подхода и сэмплирования.

- Объяснить прогноз для диагностики рака с локальной и глобальной точки зрения.

Следующие четыре раздела демонстрируют достижение вышеуказанных целей. В разделе 2.1 представлены алгоритм Isolation Forest и результаты обнаружения аномалий для набора данных рака. В разделе 2.2 предлагаются методы решения многомерных задач объяснимого ИИ, включая двухуровневый подход и сэмплирование. В разделе 2.3 результаты прогнозирования диагноза рака объясняются с использованием двухуровневого подхода и сэмплирования как с локальной, так и с глобальной точек зрения. При этом оба подхода сравниваются с моделью Xgboost проекта SHAP, которая также возвращает метки объяснения. Наконец, выводы представлены в разделе 2.4.

2.1 Система искусственного интеллекта для обнаружения рака

После рассмотрения нескольких часто используемых методов обнаружения аномалий, таких как Z-оценка (Cheadle C et al., 2003) [37], DBSCAN (Birant D et al., 2007) [34], Isolation Forest (Liu F.T et al., 2008) [40] и модифицированный алгоритм случайного леса (Tomin N et al., 2015) [46], Isolation Forest был выбран как наиболее подходящий метод для задачи диагностики рака. С одной стороны, посредством построения выборок и подвыборок Isolation Forest может значительно снизить линейную сложность по времени, таким образом уменьшая объем вычислений. С другой стороны, Isolation Forest подходит для задач, где доля аномальных данных в общем размере выборки невелика, а характеристики аномальных точек значительно отличаются от характеристик нормальных точек, что действительно имеет место быть в задачах обнаружения медицинских заболеваний.

2.1.1 Принцип изолированного леса

Алгоритм Isolation Forest основан на идее «изоляции некоторых точек из выборки данных путем рекурсивного разделения пространства выборки» (Liu F.T et al., 2008) [40]. Основываясь на «немногих» и «отличительных» свойствах выбросов, метод изоляции может быстро отделить выбросы от остальной части выборки, что является основной идеей Isolation Forest для обнаружения аномалий. На рис. 2.1 показаны детали изоляции.

Рекурсивно разбитая область изоляции рассматривается как древовидная мо-

дель. Количество разбиений (количество прямых) равно длине пути от корня до листа в модели дерева. Это означает, что длина пути выбросов будет короче, чем у нормальной точки. Из этого следует, что длина пути является мерой, характеризующей выброс, что позволяет ее использовать для идентификации выбросов.

(a) Isolating x_i (b) Isolating x_o

Рис. 2.1: Процесс выделения точки данных. (a) нормальная точка x_i изолируется за 12 шагов (т.е., требуется провести 12 прямых линий). (b) аномальная точка x_o изолируется за 4 шага; И синие, и красные области указывают на то, что каждая область имеет только одну точку данных x_i и x_o.

После добавления некоторых ограничений и параметров модель дерева становится изолированным деревом, то есть iTree. Ограничение является то, что iTree является бинарным деревом; параметры включают атрибут q, значение разделения p, каждый экземпляр x содержит атрибут q, узел, обозначенный T, T_l для левого узла и T_r - правый узел. Предполагается, что несколько экземпляров x объединяются в множество X, и множество X используется в качестве корневого узла для построения модели бинарного дерева. Среди них каждое формирование дочернего узла эквивалентно делению в пространстве выборки. Кроме того, предел высоты l iTree равен $ceiling(log_2\psi)$. После постоянной рекурсии экземпляр ψ разделяется на левую и правую стороны, и процесс останавливается до тех пор, пока не будет достигнуто условие завершения. Есть два условия завершения:

- Экземпляры неделимы.
- Высота дерева достигла предела высоты, $ceiling(log_2\psi)$.

Ограничение высоты iTree вводится с целью увеличения эффективности вычислений. Ограничение высоты устанавливается в соответствии с правилом роста средней высоты. Максимальная высота одиночного iTree определяется ψ, а

средняя высота t iTrees пропорциональна $log\psi$. При использовании iForest для обнаружения аномалий обычно обращают внимание на деревья, высота которых меньше средней высоты, а затем ищут аномальные точки рядом с корневым узлом, поэтому высота дерева задается вручную в соответствии со средней высотой Isolation Forest.

В структуре iTree все узлы, кроме внешних узлов, являются внутренними узлами (включая корневой узел). Длина пути $h(x)$ представляет собой количество ребер от корневого узла до внешних узлов, и затем может использоваться для расчета показателя аномалии. Было доказано, что длина пути аномальной точки короче, чем длина пути нормальной точки. Таким образом, iTree несет в себе информацию о том, что аномальные точки ближе к корневому узлу, а нормальные точки ближе к внешним узлам. Набор таких iTree называется собственно Isolation Forest.

2.1.2 Определения измерений: показатель аномалии

Показатель аномалии рассчитывается на основе длины пути $h(x)$, но в этом процессе есть сложная проблема (Liu F.T et al., 2012) [41]. Как упоминалось выше, максимальная высота iTree определяется ψ, но средняя высота нескольких iTree «растет» как $log\psi$. Это приводит к невозможности нормализации $h(x)$. Решение этой проблемы приходит из другой модели бинарного дерева — бинарного дерева поиска (BST), как показано в таблице 2.1.

iTree	BST
Полное бинарное дерево	Полное бинарное дерево
Выход из внешнего узла	Неудачный поиск
Случай не применим	Удачный поиск

Таблица 2.1: Список эквивалентных структур и операций в iTree и бинарном дереве поиска (BST).

Мы решаем проблему расчета оценки аномалии iForest, обращаясь к методу BST для расчета средней длины пути. Средняя длина пути на основе BST составляет:

$$c(\psi) = \begin{cases} 2H(\psi - 1) - 2(\psi - 1), & \psi > 2, \\ 1, & \psi = 2, \\ 0, & otherwise, \end{cases} \tag{2.1}$$

где H — гармоническое число, $H(a) = \ln(a)+$ константа Эйлера. $c(\psi)$ представляет собой среднее значение длины пути заданного ψ, таким образом реализуя нормализацию $h(x)$. Тогда оценка аномалии s выражается как:

$$s(x, \psi) = 2^{\frac{-E(h(x))}{c(\psi)}}. \tag{2.2}$$

Среди них $E(h(x))$ — это среднее значение соответствующей коллекции iTrees. Для оценок аномалий s существует три граничных значения, основанных на различной производительности $E(h(x))$:

- когда $E(h(x)) \to 0, s \to 1$;
- когда $E(h(x)) \to \psi - 1, s \to 0$;
- когда $E(h(x)) \to c(\psi), s \to 0.5$.

Оценка аномалии s позволяет оценить, насколько аномальным является данный экземпляр. В общем случае 1 и 0,5 являются критическими значениями:

- Если полученное значение s близко к 1, то экземпляр может быть аномальным.
- Если полученное значение s значительно меньше 0,5, то экземпляр можно смело считать нормальным.
- Если полученное значение s близко к 0,5, то это означает лишь не позволяет указать на явную аномалию экземпляра, но не означает, что аномалии нет.

Стоит отметить, что 0,5 применяется не ко всем случаям. Более качественный метод сначала определяет порог оценки путем выборочного расчета нормальных экземпляров и сравнивает оценки аномалий для обнаружения аномалий. В этих условиях порог оценки лучше соответствует распределению выборки.

2.1.3 Обнаружение аномалий с использованием Isolation Forest

После того, как iTree установлен и протестирован, для обнаружения аномалий можно применить Isolation Forest. Процесс построения iTree рассматривается как процесс обучения на подвыборках, извлеченных из набора данных, что позволяет назвать его этапом обучения.

За этапом обучения следует этап оценки. На основе созданного iTree рассчитываются показатели аномалий, то есть оценивается, насколько данная точка из выборки является аномальной по показателю аномалии.

Этап обучения

На этапе обучения пороговое значение высоты дерева l является наиболее критичным. Высота дерева должна быть ограничена примерно средней высотой дерева, поскольку средняя высота дерева определяет границу обнаружения аномалий. Точки с короткими путями (по сравнению со средним путем) имеют высокую вероятность аномалии, поэтому правильная установка высоты дерева напрямую влияет на окончательный результат обнаружения аномалии.

Алгоритм 1 показывает создание частичной модели. Построение iTree предполагает, что все экземпляры будут изолированы, а метод реализации заключается в рекурсивном делении подвыборки X'. Подвыборка X' выбирается случайным образом и не требует замены из X. Установив входные параметры, включая количество деревьев t и размер подвыборки ψ, модель Isolation Forest переходит к обучению с этими параметрами. Ограничение по высоте можно установить через размер подвыборки, затем проводить обучение по выборке, и результатом будет набор из t iTrees.

Algorithm 1: $iForest(X, t, \psi)$

Data: X - input data, t - number of trees, ψ - sub-sampling size, i - quantity of tree

Result: a set of iTrees

initialization $iForest$;

set height limit $l = ceiling(\log_2 \psi)$;

while $i \leq t$ **do**

  $X' \leftarrow sample(X, \psi)$;

  $Forest \leftarrow Forest \cup iTree(X', 0, l)$;

end

return *Forest*

Важность ψ заключается в том, что он может контролировать размер обучающей выборки. В то же время, если увеличить ψ выше определенного значения, точность обнаружения аномалий перестанет улучшаться, в то время как вычислительное время и затраты памяти возрастут. Это объясняет, почему следует использовать меньшее значение ψ. Эмпирические тесты на крупномасштабных задачах показывают, что $\psi = 2^8$ или 256 является наилучшим значением (Liu F.T. et al., 2012).) [41]. Количество деревьев t указывает на размер ансамбля. В практических экспериментах длина пути будет сходиться до $t = 100$. Следовательно, $t = 100$ является значением по умолчанию для нашего эксперимента, если не указано иное (Liu F.T. et al., 2012) [41].

В Алгоритме 2 описан процесс построения iTree. Экземпляры изолируются путем сравнения q и p. При $q < p$ экземпляр X_i будет размещен слева, а в случае $q \geq p$ он будет размещен справа. Этот процесс будет продолжаться до тех пор, пока все экземпляры не будут изолированы или высота дерева не достигнет $ceiling(log2)$.

Algorithm 2: $iTree(X, e, l)$

Data: X - input data, e - current tree height, l - height limit, Q - is the list of attributes in X

Result: an iTree

if $e \geq l$ *or* $|X| \leq 1$ **then**
  return $exNode\{Size \leftarrow |X|\}$;
  else
    randomly select an attribute $q \in Q$;
    randomly select a split point p from max and min values of attribute q in X;
    $X_l \leftarrow filter(X, q < p)$;
    $X_r \leftarrow filter(X, q \geq p)$;
    return $inNode(Left \leftarrow iTree(X_i, e+1, l)$;
      $Right \leftarrow iTree(X_r, e+1, l)$ $SplotAtt \leftarrow q, SplitValue \leftarrow p$
  end
end

Этап оценки

На этапе оценки можно обойти любой iTree внутри iForest для экземпляра X и соответственно получить ожидаемую длины пути $E(h(x))$. После этого оценка аномалии s вычисляется по $E(h(x))$. Функция $PathLength$ подсчитывает количество пройденных ребер e от начального узла до конечного узла X и возвращает $h(x)$. Если x завершается на внешнем узле с $Size > 1$, то возвращается результат $e + c(Size)$, где $c(Size)$ — корректировка, учитывающая те поддеревья, которые не были построены, потому что их высота превышает пороговую высоту дерева. Когда $h(x)$ получено, вычисляется оценка аномалии s. Детали функции $PathLength$ даны в Алгоритме 3.

Algorithm 3: $PathLength(x, T, e)$

Data: x - an instance, T - an iTree, e - current path length to be intialized to zero when first called

Result: path length of x

```
1  if T is an external node then
2  |  return e + c(T.size) {c(.) is defined in Equation 2.1}
3  end
4  a ← T.splitAtt ;
5  if x_a < T.splitValue then
6  |  return PathLength(x, T.left, e + 1);
7  |  else
8  |  |  return PathLength(x, T.right, e + 1
9  |  end
10 end
```

2.1.4 Результат обнаружения аномалии

На рисунке 2.2 показаны результаты обнаружения аномалий для набора данных о раке с использованием алгоритма Isolation Forest. О применимости модели можно судить по выходному индексу, используемому для измерения точности модели. В этой главе мы используем такие показатели, как F1-мера, TPR и FPR для измерения точности модели. Диапазон значений этих показателей составляет 0-1. Значение F1-меры и TPR, близкое к 1, означает, что точность модель выше, и наоборот для FPR. TPR = 0,6887 и FPR = 0,1261 указывают на то, что модель имеет низкий уровень ошибок диагностики и, следовательно, работает хорошо. Кроме того, F1-мера как среднее гармоническое точности и полноты широко используется для задач классификации. F1-мера = 0,7246 указывает на то, что обнаружение аномалий модели Isolation Forest имеет высокое качество.

Метод обнаружения аномалий, основанный на оценках аномалий, должен заранее определить порог оценки. При нормальном числе выборок из 569 экземпляров данных и 300 деревьев пороговое значение должно быть 0,479. На рисунке (a) экземпляры выше порога предполагаются нормальными, в противном случае – аномальными. В то же время мы утверждаем, что случаи, соответствующие показателям аномалии, превышающим 0,479 на рисунке (b), являются

аномальными данными, что означает, что у этих пациентов более вероятно наличие рака молочной железы. Результаты обнаружения аномалии подтверждают, что в некоторых случаях есть риск выявления рака. Тем не менее, будучи черным ящиком, модель Isolation Forest не способна объяснить причины формирования подобных прогнозов пользователям. Эта ситуация не допускается в медицинской диагностике, поэтому необходимо объяснить результаты работы алгоритма обнаружения аномалий, чтобы пользователи, в том числе врачи и пациенты, могли понять, как данные влияют на результаты прогнозов.

В этой главе значение Шепли используется в качестве подходящего метода объяснимого ИИ. Путем расчета значений Шепли для 30 признаков можно определить признаки, которые оказывают решающее влияние на результат прогнозирования, на основе значений Шепли. Значения Шепли предоставляются пользователям в качестве объяснений. Локальное объяснение формируется, используя значение Шепли для каждой характеристики конкретного экземпляра, и учитывает различия отдельных пациентов. Глобальное объяснение формируется на основе распределения 30 признаков для всех экземпляров и учитывает общность группы.

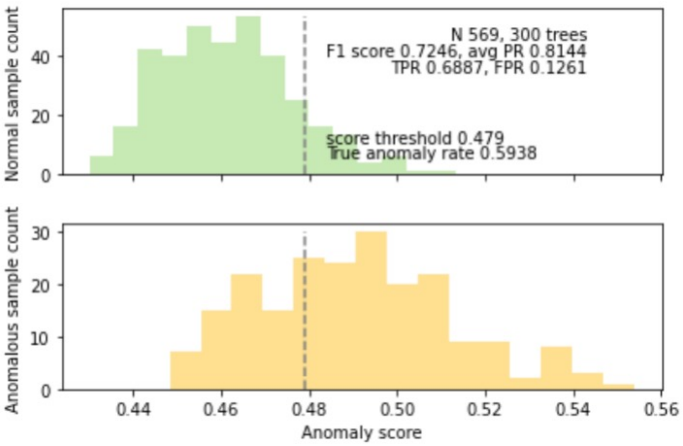

Рис. 2.2: Результат обнаружения изоляции: (а) Нормальное количество экземпляров, (б) Аномальное количество экземпляров.

Однако типичным недостатком значения Шепли является то, что оно требует большого количества вычислений по сравнению с другими методами объяснимого ИИ. В процессе медицинской диагностики задействовано несколько показателей, формирующих многомерные данные. Хотя исследователи удовлетворены точностью значения Шепли, высокие вычислительные требования неприемлемы для многомерных данных. Для решения многомерной проблемы объ-

яснимого ИИ мы предложили два решения в следующем разделе.

2.2 Объяснимое решение и алгоритмы

В настоящее время основной идеей для решения вычислительных проблем, возникающих при решении многомерных задач, является понижение размерности. Алгоритмы понижения размерности использовались в качестве критической части предварительной обработки данных, включая PCA, LDA, LLE и лапласовские собственные карты. Среди них метод главных компонент (PCA) (Wold S et al., 1987) [47] и линейный дискриминантный анализ (LDA) (Balakrishnama S et al., 1998) [32] обычно используются в качестве линейных методов понижения размерности. PCA фокусируется на сохранении деталей исходных данных, в то время как LDA ожидает, что данные смогут разделяться после уменьшения размерности. LLE (Roweis S.T et al., 2000) [45] является нелинейным алгоритмом понижения размерности, но его эффективное использование зависит от распределения исходных данных. Когда данные распределены на сфере, эффективность этого метода неудовлетворительна.

В «Объяснимом ИИ» мы используем значения Шепли для измерения вклада каждого игрока (признака) (Shapley L.S. et al., 1953) [27]. Используя значения Шепли, мы можем получить вклад каждого признака для конкретного решения (локальное объяснение) и системы в целом (глобальное объяснение). В этом разделе будет представлена комбинация значений Шепли, двухуровневого подхода и сэмплирования для решения проблемы объяснимого ИИ большой размерности.

2.2.1 Ценность Шепли и объяснимая модель

В классической теории кооперативных игр предполагается, что игроки сотрудничают для получения совместного вознаграждения, а затем совместное вознаграждение распределяется между ними (Shapley L.S. et al., 1953) [27]. Для распределения совместного вознаграждения, вводится понятие импутации. Классическим кооперативным решением или импутацией, которое мы будем использовать, является значение Шепли (Петросян О и др., 2017) [44]. Явная формула

для вычисления значения Шепли дана ниже:

$$\varphi_i = \sum_{S|i \in S \subseteq N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[v(S) - v(S \setminus \{i\})] \tag{2.3}$$

где i — игрок, N — множество всех игроков в кооперативной игре, $S \subseteq N$ — коалиция игроков, $|S|$ — количество игроков в коалиции S, $V(S)$ — характеристические функции коалиции $S \subseteq N$, задающие суммарный выигрыш игроков при формировании ими коалиции S.

Прежде чем получить значение Шепли, первым шагом должно быть получение характеристической функции $V(S)$ для каждой коалиции $S \subseteq N$. Более подробно о ее вычислении для объяснения обнаружения аномалий в системе будет рассказано в дальнейших разделах. Из уравнения (3.1) видно, что значение Шепли игрока i является взвешенной суммой вкладов игрока i для каждой коалиции S ($[v(S) - v(S \setminus i)]$). Левый множитель в произведении (3) определяет вероятность образования самой коалиции S. Чем меньше вероятность, тем меньше значение вклада игрока i в эту фиксированную коалицию S.

Используя значение Шепли, совместное вознаграждение распределяется между игроками ($\sum_{i=1}^{n} \varphi_i = V(N)$). В общем случае, если игрок вносит больший вклад в кооперацию, то его входное значение будет больше. В машинном обучении подход значения Шепли может объяснить вклад каждого значения признака. Его можно использовать для глобального объяснения и для локального объяснения.

Для применения подхода, основанного на значении Шепли, на первом шаге необходимо вычислить характеристическую функцию для каждой коалиции $S \subseteq N$, где N и S — множество и подмножество всех признаков в входных данные, связанных с диагностикой рака. Смысл характеристической функции $V(S)$ коалиции S в системе обнаружения аномалий - оценка аномалии или вероятность аномалии, рассчитанная только на основе признаков из множества S. После вычисления характеристической функции для каждой коалиции S можно вычислить значение Шепли и, как следствие, объяснить результат Isolation Forest, объяснив вклад каждого признака. Алгоритм расчета значений Шепли для объяснения Isolation Forest выглядит следующим образом:

1. Определить $n = |N|$ признаков-игроков из набора входных данных D. Для случая глобального объяснения вводом набора данных является полный

набор данных, в то время как для случая локального объяснения это один экземпляр.

2. Выбрать коалицию $S \subseteq N$ игроков-признаков. Общее количество рассматриваемых коалиций равно $2^N - 1$.

3. Запустить алгоритм Isolation Forest для каждой коалиции $S \subseteq N$, чтобы получить значение характеристической функции $V(S)$ как точность обнаружения аномалии.

4. Повторить шаги 2-3, пока не будут вычислены все значения характеристической функции $V(S)$.

5. Использовать формулу 2.3 для расчета значений Шепли для всех игроков с функциями.

2.2.2 Двухуровневый подход к многомерному объяснимому ИИ

Двухуровневый подход, разработанный в этой главе, представляет собой метод, основанный на теории кооперативных игр. Мы представляем глобальные и локальные объяснения в главе с использованием двухуровневого подхода. Этот подход возник из «двухуровневой кооперации», предложенной Н.В.Колабутиным и Л.А.Петросяном (Колабутин Н.В. и др., 2015) [25]. Этот метод делит кооперативный союз на двухуровневую структуру. Первый слой – союз участников. Второй слой рассматривает союз как участника, образуя таким образом более крупный союз. После сочетания с динамической устойчивостью формируется устойчивая «двухуровневая кооперация». Кроме того, авторы также доказали супераддитивность между двухуровневыми союзами.

Мы разработали двухуровневый подход для решения многомерных задач. Мы рассматриваем все признаки как участников совместной игры и группируем их, используя ограниченный k-means в качестве метода кластеризации с евклидовым расстоянием, затем вычисляем значение Шепли для каждой группы признаков (первый слой) и вычисляем значение Шепли для каждого признака в группе признаков (второй слой). Мы объединяем значения Шепли первого слоя и второго слоя, чтобы окончательно вычислить значение Шепли для каждого признака. Двухуровневый подход значительно повысил эффективность применения значения Шепли в случае многомерного ввода данных. Инновационный

вклад заключается в объединении традиционной теории кооперативных игр с современными методами машинного обучения для повышения производительности системы с алгоритмической точки зрения.

В случае достаточно большого числа игроков в кооперативной игре вычисление значений Шепли становится сложной задачей с вычислительной точки зрения из-за увеличения числа коалиций. Расчет характеристической функции является важным шагом для получения значений Шепли. Поэтому мы предлагаем использовать двухуровневый подход, основанный на кластеризации признаков с использованием набора исторических данных и расчетом значений Шепли для каждой группы признаков и для каждого признака в группе.

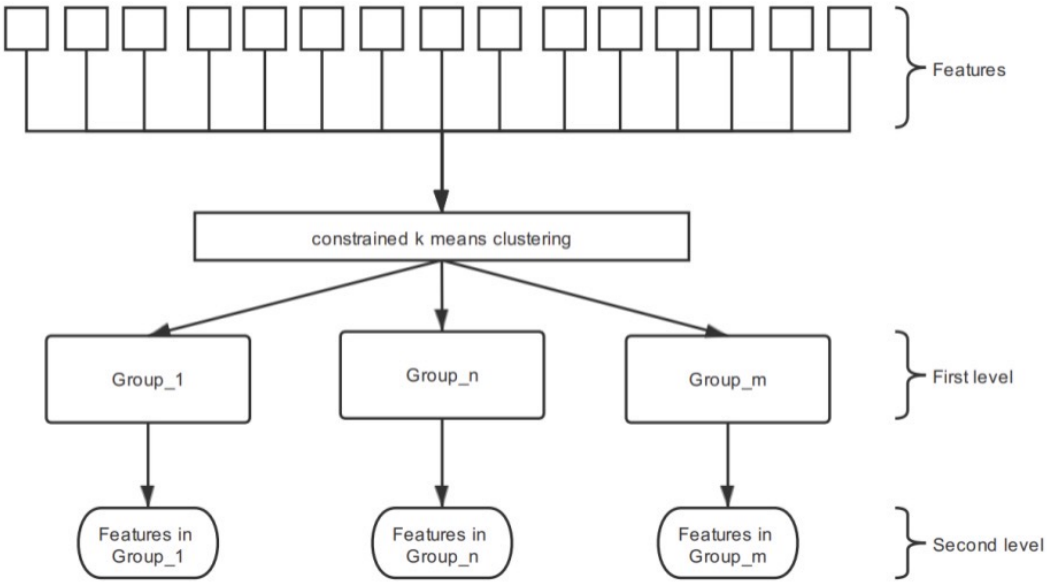

Рис. 2.3: Architecture of Bi-level approach

Нашей целью является уменьшение количество вычислений, необходимых для расчета значений Шепли, и понижение размерности является одним из эффективных методов решения этой проблемы. Для решения этой проблемы мы разработали двухуровневый подход на основе «двухуровневой кооперации». Как показано на рисунке 2.3, мы рассматриваем все признаки как n-мерные векторы и группируем их, кластеризацию с евклидовым расстоянией в качестве метрики близости и ограничением размера кластеров, чтобы сформировать несколько групп признаков в качестве первого слоя. После получения значений Шепли для каждой группы они умножаются на значения Шепли признаков в каждой группе соответственно. Наконец, рассчитывается значение Шепли для всех признаков, и именно оно используется для объяснения результатов прогно-

зирования рака. Следует подчеркнуть, что мы использовали алгоритм кластеризации k-means с евклидовой метрикой, предложенный Bradley et al. (Bradley, P.S et al., 2000) [33], чтобы сгруппировать признаки для формирования первого слоя. В двухуровневом подходе, разработанном в этой главе, размер входных данных для объяснимого алгоритма, основанного на значениях Шепли на каждом уровне, будет на приемлемом уровне для снижения вычислительной сложности. Двухуровневый подход для многомерного объяснимого ИИ формулируется следующим образом:

1. Двухуровневое структурное моделирование: определить n как количество всех признаков, то есть, размерность набора данных. Количество групп N, сформированных алгоритмом кластеризации, определяется в соответствии с доступной вычислительной мощностью, а первый слой состоит из N групп с n/N признаками в каждой группе.

2. Генерация матрицы расстояний для признаков: использовать евклидово расстояние для измерения расстояния между признаками, где размерность пространства определяется количеством 569 экземпляров данных.

3. Кластеризация по ограниченному размеру: использовать кластеризацию с ограниченным размером кластеров, чтобы разделить все признаки на N групп, где размер ограничен n.

4. Расчет значений Шепли на первом уровне: рассмотреть N групп как игроков, при этом все признаки внутри группы будут считаться одним игроком. Алгоритм Isolation Forest используется для вычисления значений Шепли для первого уровня $\phi^t[i], i = \{1...N\}$.

5. Расчет значений Шепли на втором уровне: при использовании Isolation Forest значение Шепли вычислить на втором уровне для каждой группы отдельно $\varphi_s[j], j = \{1...n\}$.

6. Расчет окончательных значений Шепли: умножить значения Шепли на первом уровне $\phi^t[i], i = \{1...N\}$ на значения Шепли на втором уровне $\varphi_s[j], j = \{1...n\}$ для вычисления окончательных значений Шепли для всех функций $\Phi[k] = \phi[i] \cdot \varphi[j], k = \{1... \cdot N\}$.

2.2.3 Подход сэмплирования для многомерного объяснимого ИИ

Второй подход к многомерному объяснимому ИИ основан на методе сэмплирования. В 2009 г. был предложен метод сэмплирования для уменьшения сложности вычисления точной формулы для значения Шепли (Фатима С.С. и др., 2005) [38]. В этой главе мы представляем глобальные и локальные объяснения обнаружения раковых аномалий с использованием сэмплирования. Фундаментальные исследования теории кооперативных игр доказали, что приближенный алгоритм Шепли эффективен для крупномасштабных игр (Castro J et al., 2009) [29], что помогает применять приближенный алгоритм Шепли к многомерным задачам объяснимого ИИ. Хотя исследователи пытались найти алгоритм, способный точно вычислить значение Шепли, ресурсы и время, затрачиваемые на это, относительно велики, а улучшение ограничено. Следовательно, имеет смысл разработать алгоритм, основанный на идее сэмплирования, который аппроксимирует значения Шепли (Castro J et al., 2009) [29]. Большинство проблем ИИ, которые необходимо решить в реальной жизни, сложны и в конечном итоге превращаются в многомерные проблемы ИИ. Следовательно, невозможно использовать точные значения Шепли для построения объяснимого ИИ до тех пор, пока не увеличится вычислительная мощность машины. Таким образом, реалистичным выбором является приближение алгоритма сэмплирования к значениям Шепли.Алгоритм представлен ниже:

1. Смоделировать игру: определить $n = |N|$ характеристику игрока из набора входных данных D. Установите размер выборки как M.

2. Задать выборку M: популяция процесса выборки P будет множеством всех возможных порядков игроков N, т. е. $P = \pi(N)$. Пусть $O : \{1, \ldots, n\} \to \{1, \ldots, n\}$ — перестановка, которая ставит в соответствие каждой позиции k игрока $O(k)$. $\pi(N)$ обозначает множество всех возможных перестановок с множеством игроков N.

3. Вычислить значение характеристической функции: характеристики, наблюдаемые в каждом экземпляре выборки, $O \in \pi(N)$, представляют собой предельные вклады игроков в порядке o, т.е. $x(o) = (x(o)_1, \ldots, x(o)_n)$, где $x(o)_i = v\left(\mathrm{Pre}^i(o) \cup \{i\}\right) - v\left(\mathrm{Pre}^i(o)\right)$.

4. Оценить значения Шепли: оценка $\hat{Sh}_i$ параметра Sh будет средним значением предельных вкладов по выборке M, т.е. $\hat{Sh} = \left(\hat{Sh}_1, \ldots, \hat{h}_1\right)$, где $\hat{Sh}_i = \frac{1}{m}\sum_{O \in M} x(o)_i$.

5. Получить окончательный результат: процесс отбора, используемый для определения выборки M, будет принимать любой порядок $O \in \pi(N)$ с вероятностью $\frac{1}{n!}$.

Используя сэмплирование, количество операций сокращается с $N^2 - 1$ до $M \cdot N$. Алгоритм выборки представлен ниже:

Algorithm 4: *AlgorithmApproShapley*

```
1  begin
2      Determine m ;
3      Cont := 0 and Sĥ_i := 0 ∀i ∈ N;
4      While Cont < m;
5      begin
6          Take O ∈ π(N) with probability 1/(n!) ;
7          For all i ∈ N;
8          begin
9              Calculate Pre^i(O);
10             Calculate x(O)_i := v(Pre^i(O) ∪ {i}) − v(Pre^i(O));
11             Sĥ_i := Sĥ_i + x(O)_i;
12         end
13         Cont := Cont + 1
14     end
15     Sĥ_i := Sĥ_i / m ∀i ∈ N
16 end
```

2.3 Результаты симуляций

Как было отмечено ранее, по контексту объекта интерпретации объяснимость подразделяется на локальную и глобальную. Локальное объяснение в основном сосредоточено на объяснении отдельных случаев. Напротив, глобальное объяснение нацелено на объяснение модели целиком. В этой главе мы демонстри-

руем как глобальные, так и локальные объяснения, используя двухуровневый подход и сэмплирование. В части локального объяснения мы также используем фреймворк SHAP для создания синтетических меток для набора данных, чтобы использовать их в качестве эталона объяснимого ИИ для измерения производительности наших подходов объяснимого ИИ.

2.3.1 Описание набора данных

В этой главе мы используем открытый набора данных из Sklearn, который называется набором данных рака молочной железы. Он включает в себя 2 класса, 30 признаков и 569 образцов. Следует отметить, что в этих признаки всего 10 основных признаков. На основе этих основных признаков были получены соответственно «среднее значение», «значение ошибки» и «наихудшее значение», что в сумме составляет 30 признаков. Метка с 0 или 1 является результатом диагностики. В этом исследовании 1 представляет пациентов с аномальными данными, то есть пациентов с риском развития рака молочной железы, в противном случае имеется в виду нормальный случай.

2.3.2 Результаты моделирования глобального объяснения: двухуровневый подход

Мы получили матрицу расстояний между различными объектами, используя евклидово расстояния для 30 признаков. Затем матрица расстояний обрабатывается ограниченного k-means с $Minsize = 5$ и $Maxsize = 10$. Результаты кластеризации групп представлены ниже:

- Группа 0: 'mean compactness', 'mean concavity', 'worst compactness', 'worst concavity', 'worst concave points', 'worst symmetry';
- Группа 1: 'mean perimeter', 'mean area', 'area error', 'worst texture', 'worst perimeter', 'worst area';
- Группа 2: 'mean smoothness', 'mean fractal dimension', 'smoothness error', 'concave points error', 'symmetry error', 'fractal dimension error';
- Группа 3: 'mean concave points', 'mean symmetry', 'compactness error', 'concavity error', 'worst smoothness', 'worst fractal dimension';

- Группа 4: 'mean radius', 'mean texture', 'radius error', 'texture error', 'perimeter error', 'worst radius'.

Результаты моделирования для первого уровня представлены на рисунке 2.4.

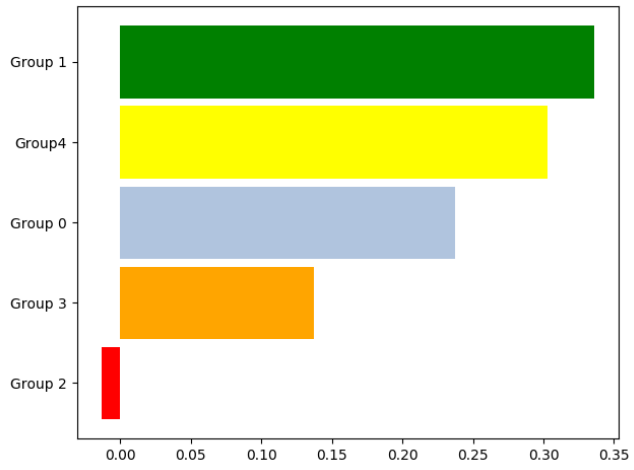

Рис. 2.4: Значение Шепли для первого (группового) уровня.

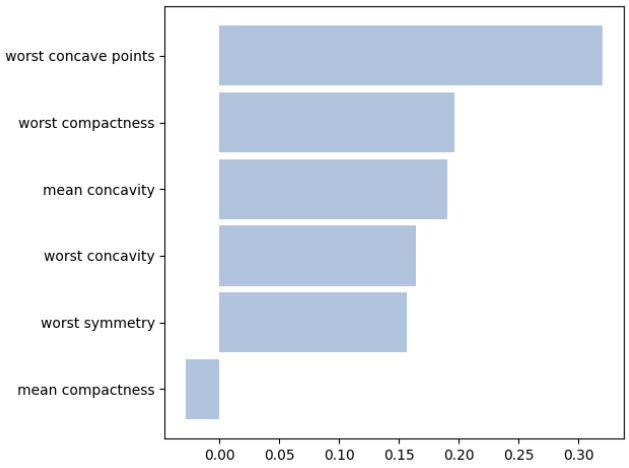

Рис. 2.5: Значение Шепли для группы 0.

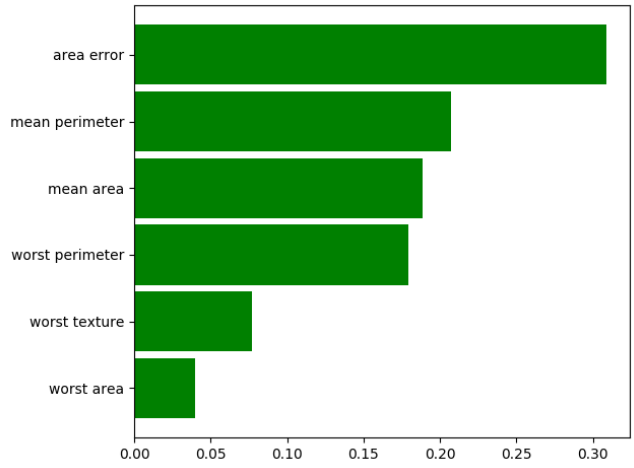

Рис. 2.6: Значение Шепли для группы 1.

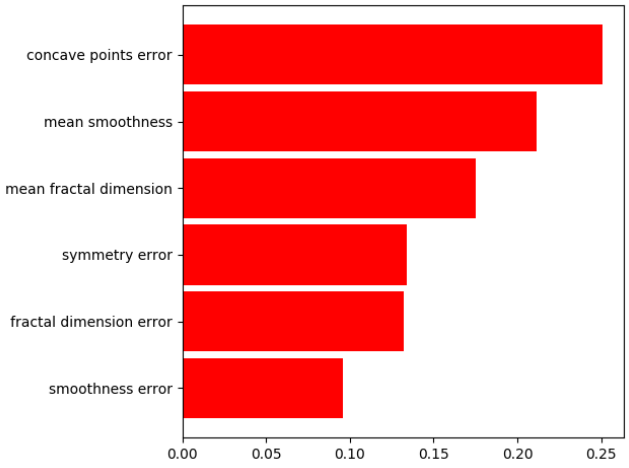

Рис. 2.7: Значение Шепли для группы 2.

Мы используем разные цвета, чтобы различать результаты объяснения групп. На оси X представлен вклад каждой группы в результаты. Группа 1 (зеленый цвет) имеет более высокий вклад (0,34). Напротив, группа 2 показывает отрицательный вклад (-0,015). Значения Шепли на первом уровне обозначим через φ^t. На первом уровне все признаки внутри группы рассматриваются как один признак-игрок (рис. 2.3).

На рисунках 2.5, 2.6, 2.7, 2.8, 2.9 показан второй уровень значения Шепли внутри каждой группы. На втором уровне игроки являются признаками из

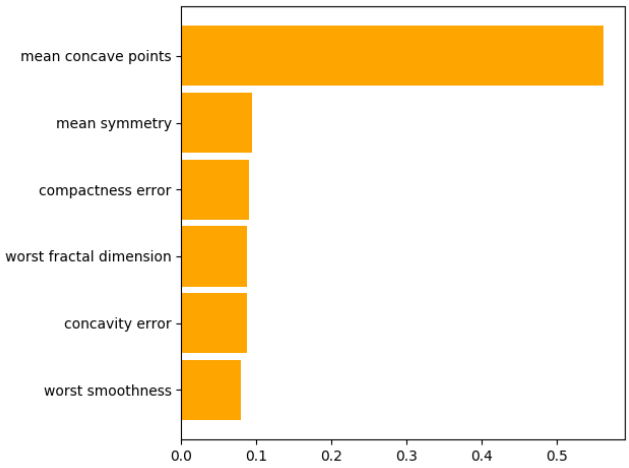

Рис. 2.8: Значение Шепли для группы 3.

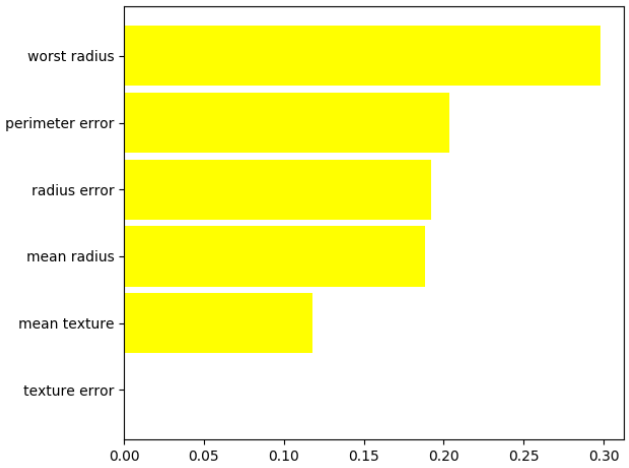

Рис. 2.9: Значение Шепли для группы 4.

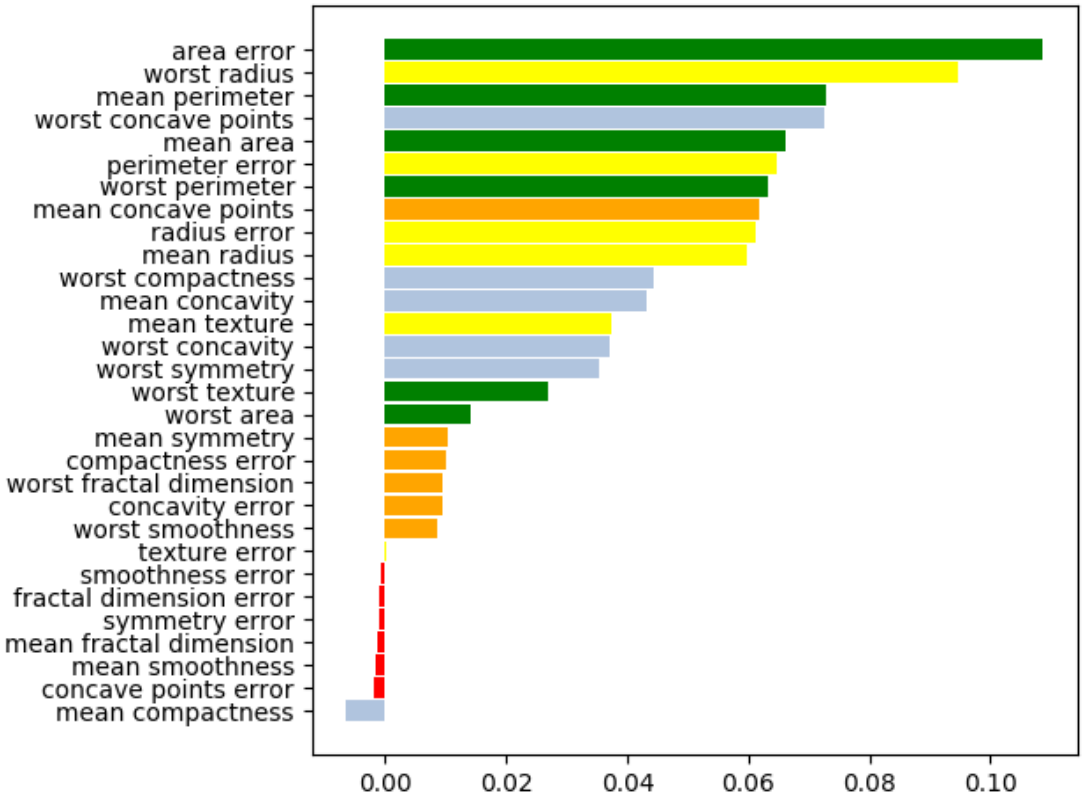

Рис. 2.10: Объяснение значения Шепли для обнаружения аномалии рака с использованием двухуровневого подхода.

одной группы. Значения Шепли второго уровня обозначим через φ_s.

На рисунке 2.10 показано окончательное объяснение значения Шепли для всех признаков, рассчитанных путем умножения первого уровня на второй: $\varphi^t \cdot \varphi_s$. На рисунке 2.10 легко увидеть, что признаки из группы 1 и группы 4 вносят значительный вклад, в то время как признаки из группы 4 имеют отрицательное значение Шепли для обнаружения раковых аномалий.

2.3.3 Результаты моделирования глобального объяснения: сэмплирование

Вместо вычисления всех характеристических функций в методе сэмплирования мы сгенерировали 300 случайных процессов (размер выборки) с 30 игроками с вероятностью $1/n!$. Каждый случайный процесс подразумевает 30 итераций для оценки характеристической функции.

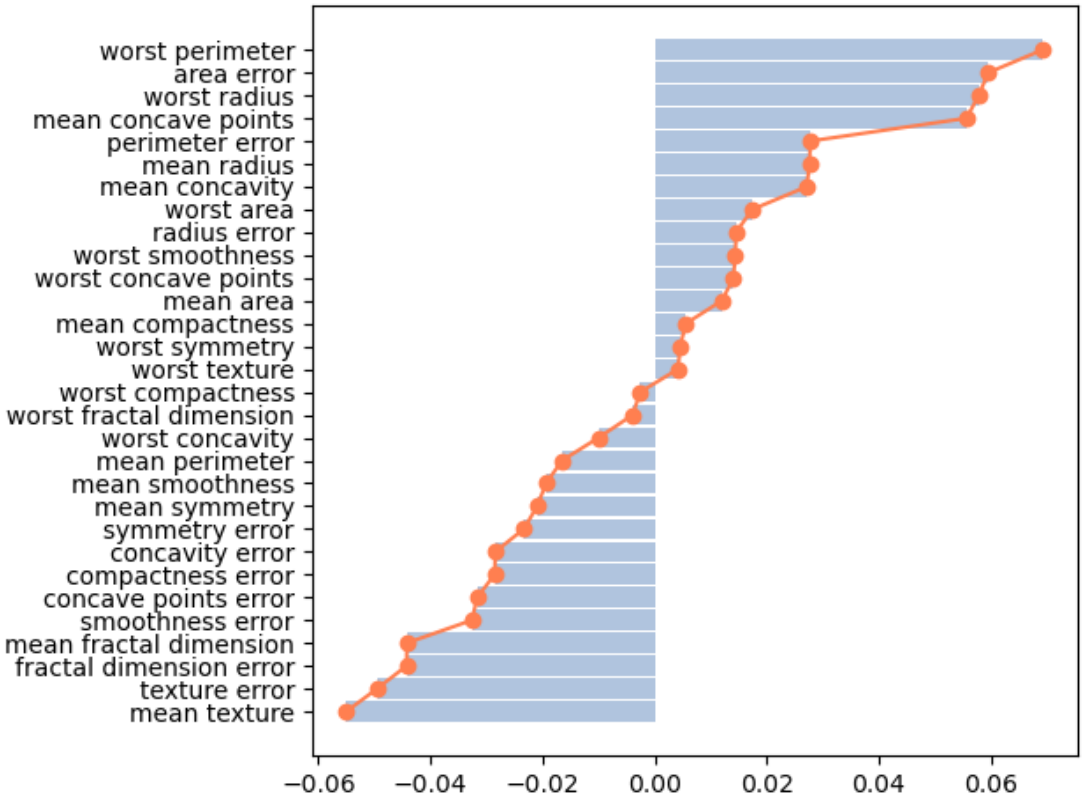

Рис. 2.11: Объяснение с помощью значений Шепли для обнаружения аномалии рака с использованием сэмплирования.

На рисунке 2.11 показаны результаты сэмплирования для объяснения с помощью значений Шепли. Из рисунков 2.10 и 2.11 легко увидеть, что метод сэмплирования дает немного другое объяснение по сравнению с двухуровневым подходом. По-прежнему сложно измерить качество объяснения. Однако мы видим, что оба объясняющих решения имеют схожий вывод:

- В основные три признака в двухуровневом режиме и сэмплировании входят “area error” и “worst radius”. Неточность может быть вызвана кластеризацией в двухуровневом подходе и аппроксимацией в методе сэмплирования. Признаки, находящиеся на местах с 4 по 10, почти идентичны для обоих

подходов.

- Имеются объекты с отрицательными объясняющими значениями. Отрицательные значения Шепли получаются, когда точность предсказания с коалицией признаков A меньше, чем с совокупностью признаков B ($coalitionA \in coalitionB$). При обнаружении рака это происходит, когда конкретное значение изменяется при вычислении точности прогноза. Нетрудно заметить, что в результатах объяснения при сэмплировании количество признаков с отрицательными объясняющими значениями в два раза больше, чем при двухуровневом подходе. Причина вызвана уменьшением размерности в двухуровневом подходе.
- Оба подхода показывают, что признаками с наихудшими отрицательными объясняющими значениями являются “mean compactness”, “concave points error”, “fractal dimension error”, “mean fractal dimension” и т.д.

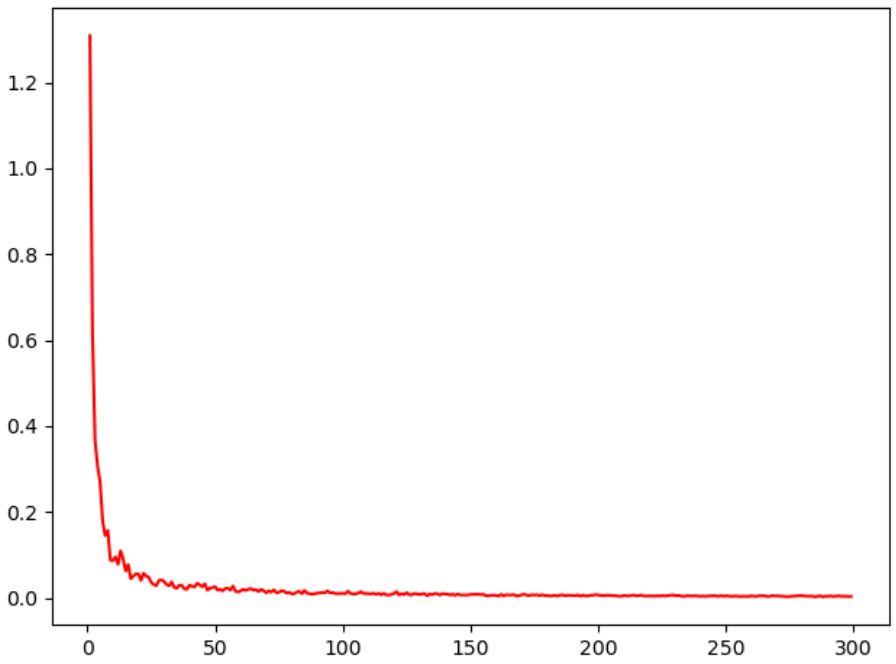

Рис. 2.12: Сходимость значения Шепли для метода сэмплирования.

В этой главе размерность набора входных данных составляет 30 признаков, поэтому размерность системы объяснимого ИИ составляет 30 признаков-игроков. Было бы затруднительно вычислить значения Шепли, используя точный подход, перечисляя и анализируя характеристическую функцию для всех

коалиций. Поэтому мы предложили использовать метод сэмплирования для аппроксимации результатов точного подхода. Из рисунка 2.12 легко увидеть, что после 100-150 итераций метод выборки сходится, а значения Шепли существенно не меняются. Это означает, что мы можем использовать соответствующие значения Шепли для объяснения обнаружения рака. Количество итераций для вычисления значений, объясняющих Шепли, при методе выборки увеличено по сравнению с точным подходом, с $30^2 - 1$ до $100 \cdot 30$ итераций.

Ошибка рассчитывается по формуле $\sum(sh_i^{k-1} - sh_i^k)$, где k — номер итерации, а i — индекс признака. Процедура останавливается, когда изменение значения Шепли при переходе к следующей итерации близко к 0.

2.3.4 Локальное объяснение: сравнение двухуровневого подхода с сэмплированием

Чтобы измерить эффективность объяснения двух предложенных нами подходов, мы используем классический метод Xgboost из проекта SHAP. Xgboost используется для создания меток объяснимого ИИ для локального объяснения, а не для сравнения качества объяснения с проектом SHAP. Сравнить наши подходы с проектом SHAP невозможно, поскольку проект SHAP не содержит модуля обнаружения аномалий Isolation Forest. В таблице 2.2 мы можем увидеть значения характеристик рака и результаты объяснения с использованием двухуровневого подхода, сэмплирования и меток объяснимого ИИ, полученных с использованием Xgboost (проект SHAP) для конкретного пациента, больного раком.

Из таблицы 2.2 легко увидеть, что:

- Двухуровневый подход показывает, что “mean compactness”, “worst compactness”, “mean concavity”, “mean concave points”, “mean fractal dimension”, “worst perimeter”, “worst concavity” и т. д. важны для прогнозирования рака.
- Сэмплирование показывает, что “perimeter error”, “texture error”, “mean concavity”, “mean concave points”, “mean fractal dimension”, “worst perimeter”, “worst concavity” вносят наибольший вклад в прогнозирование рака с помощью Isolation Forest для данного конкретного пациента.
- Оба подхода предполагают, что “mean concavity”, “mean concave points”,

Features	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error	perimeter error	area error	smoothness error	compactness error	concavity error	concave points error	symmetry error	fractal dimension error	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
Input data	17.99	10.38	122.8	1001.0	0.1184	0.2776	0.3001	0.1471	0.2419	0.0787	1.095	0.9053	8.589	153.4	0.0064	0.049	0.0537	0.0159	0.03	0.0062	25.38	17.33	184.6	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
Bi-level	0.03019	0.03282	0.03118	0.03498	0.03146	0.04157	0.04068	0.0342	0.02757	0.03023	0.03282	0.03282	0.03282	0.03498	0.03023	0.02757	0.02757	0.03023	0.03023	0.03023	0.03282	0.03498	0.03498	0.03498	0.02757	0.04068	0.04068	0.04068	0.04068	0.02757
Sampling	-0.00127	-0.0001	0.00337	-0.00243	-0.0009	0.00238	0.00336	0.00406	0.00151	0.00724	-0.00024	0.00419	0.00524	-0.00157	-0.00565	-0.00252	-0.0014	-0.00704	0.0024	-0.00585	-0.00354	-0.00181	0.00752	-0.00286	-0.00399	0.00074	0.00561	-0.0063	-0.0046	0.00443
XAI-labels (Xgboost)	-0.00909	-0.02957	0.0	0.0	-0.00113	-0.00051	0.0085	0.0963	0.00005	0.0	0.09408	0.00241	0.00049	0.00977	-0.00014	0.0	0.0	0.0	0.0	0.00049	0.19534	-0.06579	0.0	0.01275	0.00054	0.0	0.00301	0.06082	0.0	0.0

Таблица 2.2: Результаты объяснения для конкретного пациента, больного раком, полученные с помощью двухуровневого подхода, сэмплирования и меток объяснимого ИИ, полученных через Xgboost.

"mean fractal dimension", "worst perimeter", "worst concavity" вносят значительный вклад в прогнозирование рака в Isolation Forest.

- Игроки с наихудшим вкладом в двухуровневом подходе также плохо работают в сэмплировании, где большинство из них показывают отрицательное значение.

В настоящее время в области объяснимого ИИ нет общего показателя эффективности для измерения результатов объяснения. Мы предлагаем оценить качество объяснимого ИИ, используя внутренние метки для каждого признака, предполагая, что они являются истинной причиной аномалии или положительным прогнозом рака для пациента. Поскольку ни одна метка не может быть использована в качестве метки объяснимого ИИ в нашем наборе входных данных, мы решили использовать Xgboost для создания синтетической метки по истинной причине аномалии. Одна из наших будущих целей — разработать эталонный набор данных с метками как для аномалии, так и для причины аномалии (метка объяснимого ИИ). Тем не менее, в настоящее время нам необходимо использовать этот метод, чтобы представить потенциальный алгоритм оценки качества объяснимого ИИ. Из таблицы 2 легко проанализировать ошибки между синтетической меткой (Xgboost в рамках SHAP) и предлагаемыми подходами (сэмплирование и двухуровневый подход). Квадратичная ошибка или произво-

дительность объяснимого ИИ для метода сэмплирования составляет 0,06725, а для двухуровневого подхода — 0,07325. Квадратичная ошибка вычисляется по формуле $\sum_{i=0}^{N}(sh_i^{propose} - sh_i^{benchmark})^2$. Кажется, что сэмплирование имеет лучшее объяснение, чем двухуровневый подход. Но важно отметить, что двухуровневый подход можно комбинировать с сэмплированием для решения задачи очень большой размерности и даже можно комбинировать с существующими проектами, такими как SHAP.

2.4 Заключение к главе 2

1. Результаты глобального объяснения показывают, что ошибка площади (area error), наихудший радиус (worst radius), средняя площадь (mean area), ошибка периметра (perimeter error) и средние вогнутые точки (mean concave points) вносят наиболее значительный вклад в обнаружение рака обученной моделью.

2. Результаты локального объяснения показывают, что средняя вогнутость (mean concavity), средние вогнутые точки (mean concave points), средняя фрактальная размерность (mean fractal dimension), наихудший периметр (worst perimeter), наихудшая вогнутость (worst concavity) являются основными причинами прогнозирования подозрения на рак у данного экземпляра (пациента) с помощью алгоритма Isolation Forest.

3. Метод сэмплирования точнее, чем двухуровневый подход, основанный на синтетических метках, полученных из Xgboost (проект SHAP). Но двухуровневая система может быть применена к другим решениям объяснимого ИИ для решения очень сложной задачи объяснимого ИИ.

Глава 3

Объяснимый искусственный интеллект: подход к сэмплированию на основе графа для многомерной системы искусственного интеллекта

В этой главе мы представим наш метод, называемый методом сэмплирования на основе графов. Такой подход способствует ускорению вычислений за счет использования взаимосвязей между всеми игроками. В разделе 3.1 дается краткий обзор литературы и определений, включая алгоритм Isolation Forest, значение Шепли, метод сэмплирования Шепли (большинство из которых были представлены в главах 1 и 2). В разделе 3.2 вводятся наши ключевые методы, включая генерацию карты взаимосвязей, алгоритм случайного поиска и измерения сходимости. В разделе 3.3 описывается эксперимент: набор данных, система ИИ, наше решение и результаты. В разделе 3.4 представляются наши выводы и обсуждение будущей работы.

Представленные нами новые методы помогли снизить стоимость расчетов примерно на 40% на основе наборов данных тестов на рак. Теоретически наш подход может быть применен ко всем приложениям, использующим значения Шепли. Новизна этого исследования заключается в нашем предложении использовать графический метод для описания взаимосвязей между всеми игроками и использовать эту информацию, чтобы помочь пользователям собрать больше полезной информации для создания сэмплирования.

3.1 Объяснимая система искусственного интеллекта для обнаружения рака

3.1.1 Isolation Forest и обнаружение рака

Isolation Forest - это эффективный алгоритм обнаружения аномалий, который создает несколько деревьев решений (называемые изолирующими деревьями) для выявления аномалий в наборе данных. Алгоритм использует тот простой факт, что аномалии, как правило, редки и независимы от большинства других точек в пространстве признаков, что приводит к сокращению длины l-образного пути, необходимого для изоляции этих аномалий от других точек. По сравнению с другими методами обнаружения аномалий Isolation Forest обладает меньшей вычислительной сложностью и лучшей производительностью [40].

В нашей работе мы использовали алгоритм Isolation Forest для выявления рака. Набор данных получен из Sklearn и содержит две категории, 30 признаков и информацию о 569 пациентах. В этом наборе данных: 1 указывает на то, что у пациента рак, в то время как 0 означает, что пациент в норме. На следующем рисунке показано использование алгоритма Isolation Forest [60].

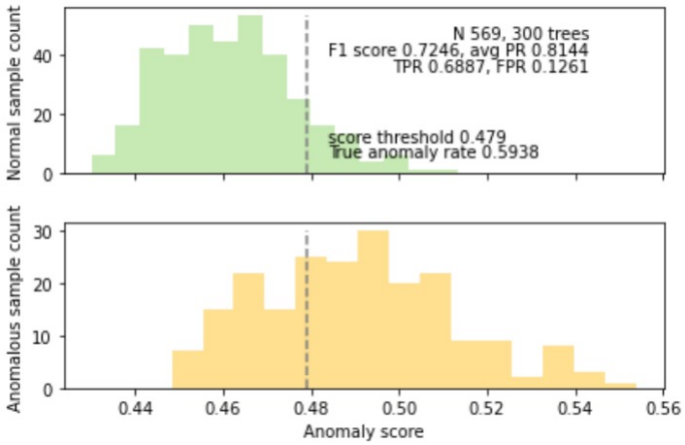

Рис. 3.1: Результат обнаружения изоляции. a) нормальный подсчет проб; (b) Аномальное количество образцов.

3.1.2 Значение Шепли

Значение Шепли [27], происходящее из теории игр, было предложено Шепли Л. в 1953 году. Оно используется для решения проблемы распределения вклада в кооперативных играх, обеспечивая справедливое измерение предельного

вклада каждого игрока в коалиции. Значение Шепли определяется следующим уравнением:

$$\varphi_i = \sum_{S|i \in S \subseteq N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[v(S) - v(S \setminus \{i\})] \tag{3.1}$$

В 2017 году Скотт и его группа предложили применить значение Шепли к области объяснимого ИИ для объяснения моделей прогнозирования [12]. Свойства эффективности, симметрии, фиктивности и аддитивности позволяют значению Шепли давать достоверное объяснение решениям. Однако вычислительные затраты для получения значения Шепли неприемлемы для приложений. Некоторые исследователи предложили использовать методы сэмплирования или аппроксимации для эффективного вычисления значения Шепли, такие как метод сэмплирования Шепли.

3.1.3 Подход Шепли к сэмплированию

В 2009 году Кастро Дж. [29] предложил метод сэмплирования для уменьшения вычислительных затрат значения Шепли. Этот эффективный метод оценки основан на случайном сэмплировании и в первую очередь решает проблему расчета с большим количеством признаков. Он аппроксимирует значение Шепли путем построения множества случайных перестановок признаков и вычисления предельного вклада каждого признака, тем самым снижая вычислительную сложность. Такой подход снижает затраты времени и памяти, необходимые для вычисления значения Шепли, делая весь процесс вычислений более эффективным за счет случайного сэмплирования. Ниже приведен псевдокод этого алгоритма:

Algorithm 5: *AlgorithmApproShapley*

begin
  Determine m ;
  $Cont := 0$ and $\hat{Sh}_i := 0 \; \forall i \in N$;
  While $Cont < m$;
  begin
    Take $O \in \pi(N)$ with probability $1/(n!)$;
    For all $i \in N$;
    begin
      Calculate $Pre^i(O)$;
      Calculate $x(O)_i := v(Pre^i(O) \cup \{i\}) - v(Pre^i(O))$;
      $\hat{Sh}_i := \hat{Sh}_i + x(O)_i$;
    end
    $Cont := Cont + 1$
  end
  $\hat{Sh}_i := \frac{\hat{Sh}_i}{m} \; \forall i \in N$
end

3.1.4 Результаты и анализ

В предыдущей главе ошибка вычислялась по формуле $\sum(sh_i^{k-1} - sh_i^k)$, где k — итерация, а i — индекс признака. Можно заметить, что алгоритм обладает быстрой сходимостью в пределах наблюдения в 300 итераций (рисунок 3.2 и рисунок 3.3). С численной точки зрения, даже когда ошибка между sh_i^{k-1} и sh_i^k сходится, этого может быть недостаточно для получения информации о ранге. В некоторых случаях отрицательные значения также могут привести к менее точным результатам. Чтобы решить эту проблему, мы предлагаем использовать как среднюю абсолютную ошибку (MAE), так и ранговую корреляцию Спирмена для измерения сходимости результатов.

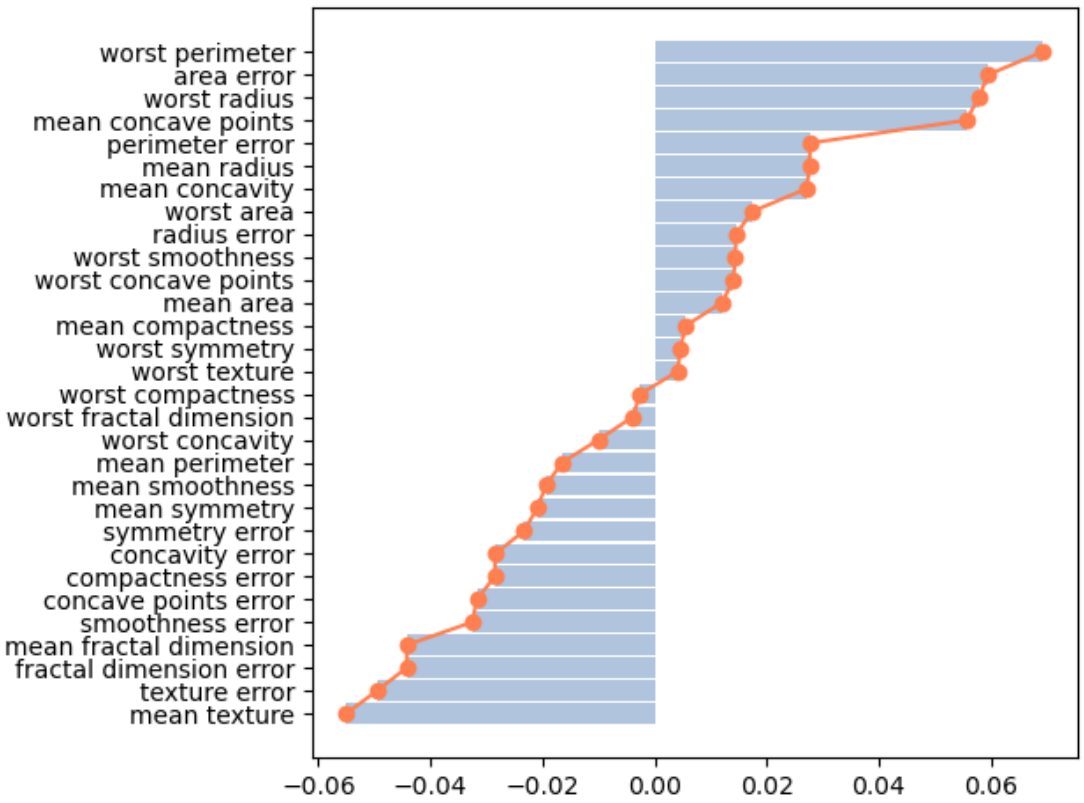

Рис. 3.2: Объяснение значения Шепли для обнаружения аномалии рака с использованием подхода сэмплирования.

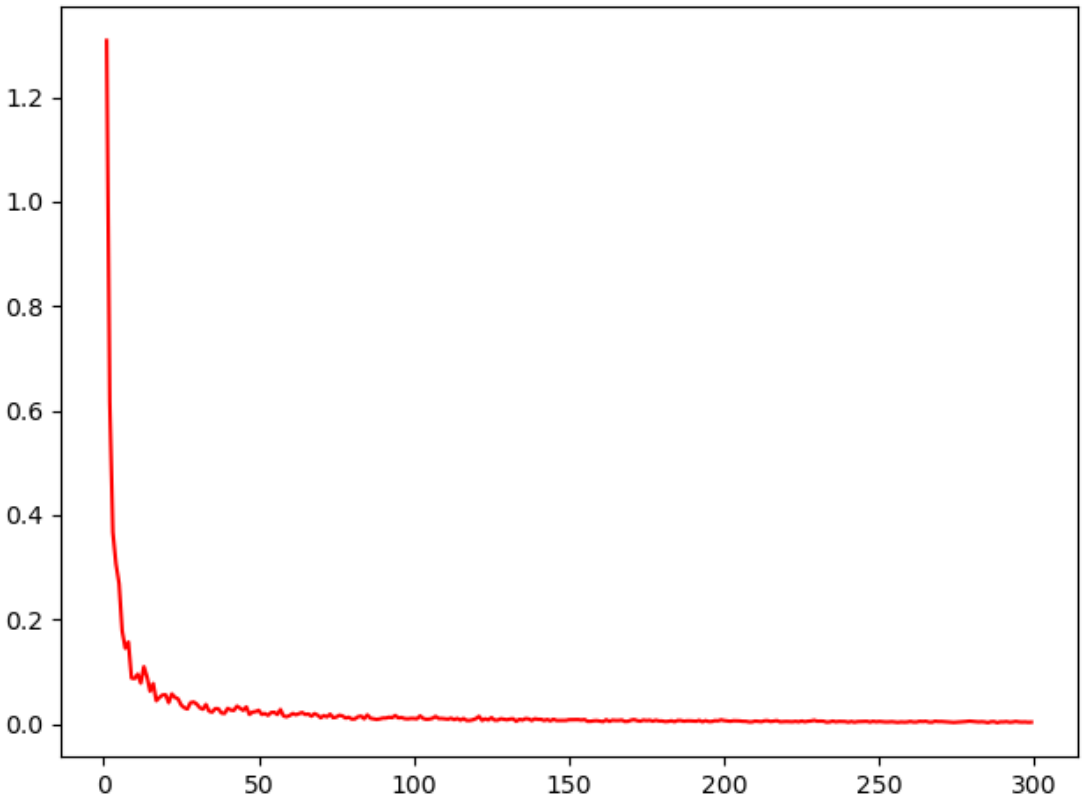

Рис. 3.3: Сходимость алгоритма сэмплирования Шепли

3.2 Карта взаимосвязей и сэмплирование на основе смещенного графа

3.2.1 Обзор алгоритма

Алгоритм, как правило, состоит из трех шагов. Первый шаг включает в себя создание карты взаимосвязей на основе коэффициента корреляции Пирсона, который подробно описан в разделе 3.2. Карта взаимосвязей будет использоваться для создания графа. Второй шаг предполагает создание случайных путей на основе сэмплирования графа, как описано в разделе 3.3. Сгенерированный случайный путь будет использоваться в методе сэмплирования Шепли. И, наконец, третий этап касается метода сэмплирования Шепли, более детальная информация о котором представлена в разделе 3.1.3.

Algorithm 6: Graph Based Sampling Approach for the Shapley value

Data: Data D, filtering parameter p, sample size M, sample length L_j, sample size N

Result: $\hat{Sh}_i$

Initialize $c \leftarrow 0$ $\hat{Sh}_i \leftarrow 0$

$a_{ij} \leftarrow \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$, $a_{ij} \in A$

For all $a \in A$ **with** $a < p$ **do** $a \leftarrow 0$

while $c < M$ **do**

- $l = 0$
- random select $player_i, L$, let O = O $\cup$ $player_i$
- **while** $l < L$ **do**
  - Random select next $player_j$ by probability $P \sim \frac{A_{ij}}{\sum A_i}$
  - Add $player_j$ to the set: $O = O \cup player_j$
  - $i = j$, $l = l + 1$
- **end**
- **for** i *in* O **do**
  - Calculate $Pre^i(O)$
  - Calculate $x(O)_i = v(Pre^i(O) \setminus \{i\}) - v(Pre^i(O))$
  - $\hat{Sh}_i = x(O)_i$
  - $count_i = count_i + 1$
- **end**
- $c = c + 1$

end

Calculate $\hat{Sh}_i = \hat{Sh}_i / counter_i$

3.2.2 Коэффициент корреляции Пирсона

Коэффициент корреляции Пирсона — это метод, используемый для исследования линейной зависимости между двумя непрерывными переменными [61]. Этот коэффициент использовался в различных медицинских учреждениях. Например, в корреляционном анализе биомаркеров он измеряет степень корреляции между различными биомаркерами, помогая выявить потенциальные факторы риска или патологические механизмы [62]. В другом случае исследования корре-

ляции экспрессии генов могут использовать коэффициент корреляции Пирсона для выявления паттернов совместной экспрессии, функционального сходства и регуляторных механизмов между генами [63].

В нашем исследовании мы пытаемся объяснить влияние различных данных на прогнозирование исходов рака на основе медицинских физических показателей с помощью объяснимой технологии искусственного интеллекта. Поэтому мы используем коэффициент корреляции продукта Пирсона и момента, чтобы исследовать взаимосвязь между различными объектами и построить карту взаимосвязей между всеми объектами. Уравнение для коэффициента корреляции произведения Пирсона и момента приведено в уравнении 3.2.

В нашем исследовании мы пытаемся объяснить влияние различных данных на прогнозирование рака на основе медицинских физических показателей с помощью объяснимой технологии искусственного интеллекта. Поэтому мы используем коэффициент корреляции Пирсона, чтобы исследовать взаимосвязь между различными функциями и построить карту взаимосвязей между всеми функциями. Коэффициента корреляции Пирсона задается уравнением 3.2.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.2}$$

где:

- x_i и y_i — наблюдаемые значения переменных X и Y.
- $\bar{x}$ и $\bar{y}$ - средние значения переменных X и Y.
- n — количество данных точек.

Рисунок 3.4 иллюстрирует полностью связанную карту взаимосвязей между всеми объектами. Мы генерируем веса ссылок, используя уравнение 3.2, и сохраняем их в матрице смежности. Для уменьшения связей предлагается фильтрация слабых связей путем нахождения квартиля q%, который можно корректировать в соответствии с требованиями эффективности. Такой подход позволяет уменьшить сложность графа, сохранив при этом все важные связи между игроками, как показано на рисунке 3.5.

3.2.3 Метод предвзятого случайного поиска пути

Чистое случайное сэмплирование со всеми игроками, используемая в оригинальном аппроксимирующем алгоритме Шепли 5, по-прежнему сталкивается с

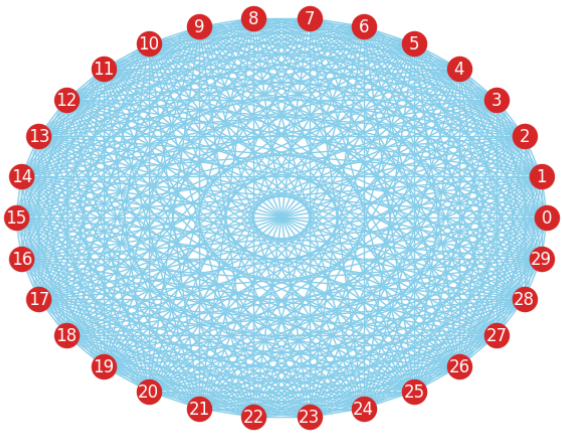

Рис. 3.4: Оригинальная карта взаимосвязей

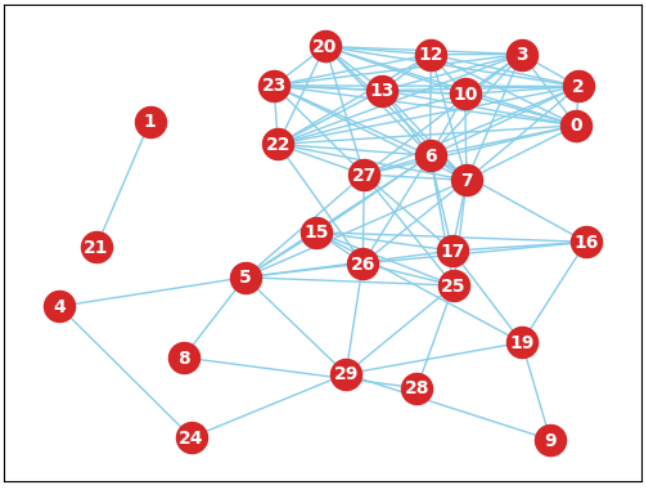

Рис. 3.5: Карта взаимосвязей с самой важной ссылкой

проблемой высоких вычислительных затрат. Мы стремимся разработать более быстрый метод расчета, используя карту взаимосвязей в качестве образцов. Этот подход предполагает создание небольших коалиций при сохранении надежной информации о взаимодействии между игроками. Алгоритм для метода поиска смещенного случайного пути показан в Algorithm 7.

Чисто случайное сэмплирование со всеми игроками, используемая в исходном аппроксимирующем алгоритме Шепли 5, по-прежнему сталкивается с проблемой высоких вычислительных затрат. Мы стремимся разработать более быстрый метод расчета, используя карту взаимосвязей в качестве образцов. Этот подход предполагает создание меньших коалиций при сохранении надежной информации о взаимодействии между игроками. Алгоритм 7 иллюстрирует метод поиска смещенного случайного пути.

Algorithm 7: Random Path Generation

Input: Number of sample paths N, random length L, adjacency matrix A

Output: N generated paths

for $i \leftarrow 1$ **to** N **do**
  Select a random starting node $Node_1^{(i)}$
  for $t \leftarrow 1$ **to** $L-1$ **do**
    Compute the sum of weights for adjacent nodes: $S_t^{(i)} = \sum_k A_{Node_t^{(i)},k}$
    Normalize the weights in the adjacency matrix row: $P_t^{(i)} = \frac{A_{Node_t^{(i)}}}{S_t^{(i)}}$
    Choose the next node based on normalized weights: $Node_{t+1}^{(i)} \sim P_t^{(i)}$
  end
end

3.2.4 Измерения для улучшения сходимости

Как упоминалось в разделе 3.1.4, числовая сумма ошибок неадекватно отражает стабильность результатов. Мы предлагаем использовать как среднюю абсолютную ошибку (MAE), показанную в уравнении 3.3, так и ранговую корреляцию Спирмена, представленную в уравнении 3.4, чтобы определить, сошлись ли результаты. MAE помогает нам избежать шума от суммы ошибок между положительными и отрицательными факторами, в то время как ранговая корреляция Спирмена помогает отслеживать информацию о ранжировании.

Уравнение средней абсолютной ошибки (MAE) определяется следующим образом:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.3}$$

где,

- n - количество наблюдений;
- y_i - истинное значение i-го наблюдения;
- $\hat{y}_i$ - прогнозируемое значение i-го наблюдения.

Уравнение коэффициента ранговой корреляции Спирмена задается формулой:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{3.4}$$

где,

- ρ - коэффициент ранговой корреляции Спирмена;
- d_i - разницу между рангами парных точек данных для i-го наблюдения;
- n - количество наблюдений (пар данных);
- $\sum d_i^2$ – сумма квадратов разностей между рангами парных точек данных.

3.3 Результаты моделирования

3.3.1 Описание набора данных

В этой главе мы используем хорошо известный набор данных о раке молочной железы с открытым исходным кодом от Sklearn. Этот набор данных состоит из 2 классов, 30 признаков и 569 образцов. Важно отметить, что среди этих признаков есть, по сути, 10 основных. На их основе были рассчитаны три различных значения - «среднее значение», «значение ошибки» и «наихудшее значение», в результате чего было получено в общей сложности 30 признаков. Диагностический результат представлен меткой 0 или 1, где 1 означает пациентов с аномальными данными, указывающими на риск развития рака молочной железы, а 0 означает нормальные результаты.

3.3.2 Генерация и конфигурация карты взаимосвязей

Полносвязанная карта строится на основе коэффициента корреляции Пирсона, представленного в разделе 3.1. Результат показан на рисунке 3.6. После построения карты мы фильтруем слабые связи между игроками, устанавливая параметр p для фильтрации слабых связей как квартиль 75%. Результат показан на рисунке 3.7.

Хотя теоретически слабые связи оказывают незначительное влияние на на союзы или концепции теории графов, мы все же хотим оценить влияние этой операции на результаты. С точки зрения объяснимого ИИ, пользователи обычно больше заботятся о самых сильных или самых слабых участниках. Поэтому мы разделили рейтинг вклада на интервалы и оценили точность, сравнив частоту попаданий участников в разные интервалы с базовым рангом. Мы предлагаем два метода оценки: 1. Интервалы Top10, Mid10 и Bot10 оценивают сильные,

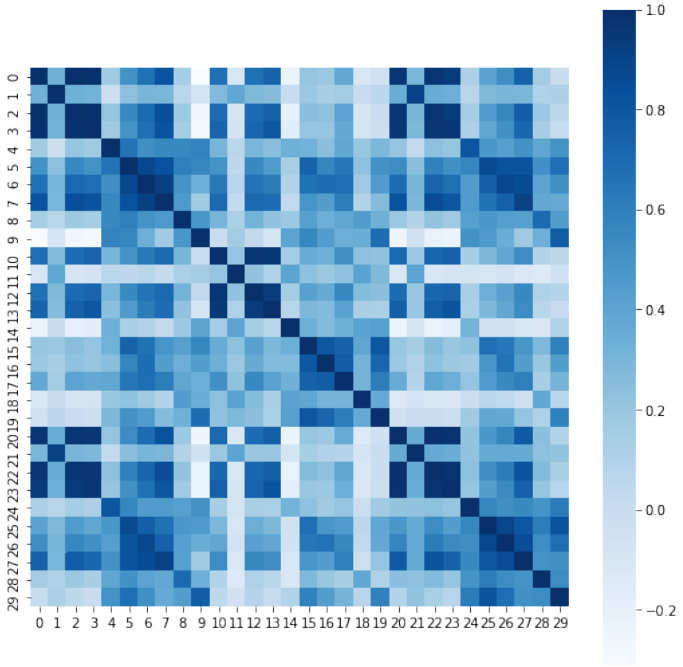

Рис. 3.6: Взаимодействие на основе корреляции

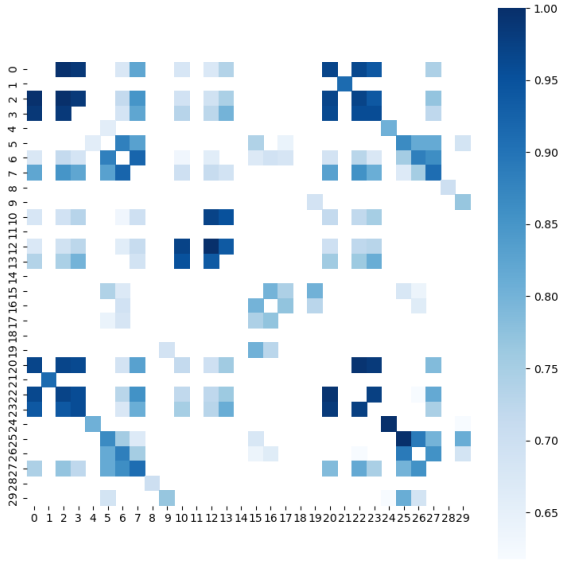

Рис. 3.7: После снижения отношений на 75%

средние и низкие участников; 2. Положительные и отрицательные интервалы оценивают положительных и отрицательных участников. На рисунке 3.8 показана взвешенная комплексная точность в сочетании с этими двумя методами оценки, суммированная с весовым отношением 3 к 7. На рисунке 3.9 показано попадания в интервалы положительного и отрицательного вклада. Из этих двух графиков сходимости видно, что параметр фильтрации p% мало влияет на точность больших интервалов. Другими словами, фильтрация слабых эффектов не вызывает большого изменения в результатах, но может привести к нестабильности в процессе итерации. Таким образом, мы определяем $p = 75\%$ для анализа нашего алгоритма в следующем разделе. Кроме того, сходимость здесь относится к частоте попаданий в разные интервалы. В процессе расчета значений Шепли внутреннее ранжирование и значения Шепли для разных интервалов все еще колеблются в зависимости от сэмплирования. Подробнее см. в разделе 3.3.3. Кроме того, сэмплирование на основе смещенного графа генерируются с использованием метода генерации случайного пути с использованием алгоритма 7. Чтобы гарантировать стабильность и достоверность результатов, мы предлагаем две дополнительные настройки для наших экспериментов:

Сэмплирование на основе смещенного графа генерируется с использованием метода генерации случайных путей с использованием алгоритма 7. И чтобы

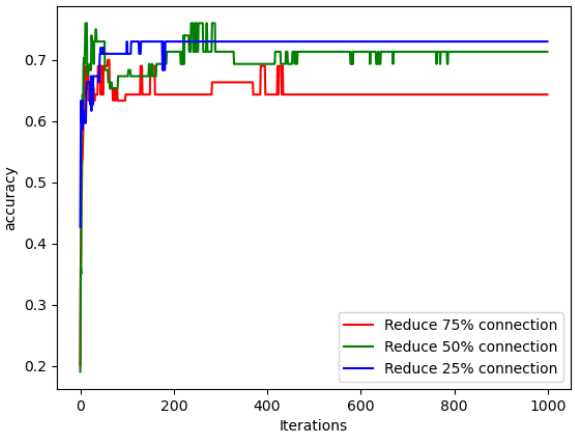

Рис. 3.8: Точность для смешанных интервалов

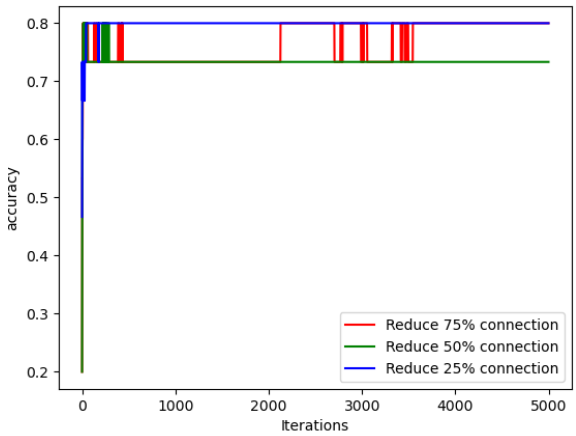

Рис. 3.9: Точность для отр. и пол. интервалов

гарантировать стабильность и достоверность результатов, мы предлагаем три дополнительные настройки для наших экспериментов:

- Присвоение нулевого значения неучастникам. Наиболее часто используемые базовые значения: ноль, среднее значение или значение, основанное на распределении.
- Сохранение значений NaN при использовании метода сэмплирования Шепли. Вместо того, чтобы с самого начала заполнять среднее или нулевое значение для отсутствующих игроков, мы сохраняем значение NaN, чтобы предотвратить внесение нестабильности и шума в окончательные результаты.
- Исключение значения Шепли первого игрока при использовании метода сэмплирования Шепли. В нашем исследовании мы рассматриваем точность как функцию выигрыша, когда в игре имеет значение порядок игроков. Исключение первого игрока поможет нам получить более честный результат.

3.3.3 Анализ результатов

Основываясь на измерениях корреляции MAE и индекса Спирмена, мы протестировали как исходный метод сэмплирования Шепли, так и метод сэмплирования Шепли на основе графа с 10 000 итераций. Как видно на рисунках 3.10 и 3.11, MAE стабилизируется после 300 итераций, а корреляция Спирмена стабилизируется примерно через 5000 итераций. Существуют колебания вблизи

7500-й итерации, вызванные случайным сэмплированием, однако значения ранга и Шепли остаются почти одинаковыми между 5000-й и 10000-й итерациями для исходного метода сэмплирования.

Что касается метода сэмплирования графа, то здесь ситуация аналогичная. Колебания между 5000-й и 10000-й итерациями происходят чаще из-за случайных размеров сэмплирования. Ранги немного меняются: только два признака опустились на одну позицию в 10000-й итерации по сравнению с 5000-й итерацией.

Для объективного наблюдения мы можем обоснованно заключить, что результаты стабилизируются после 5000 итераций. Применяя подход графового сэмплирования Шепли с тем же количеством итераций, сквозные временные затраты сокращаются на 40% благодаря дизайну со случайным размером.

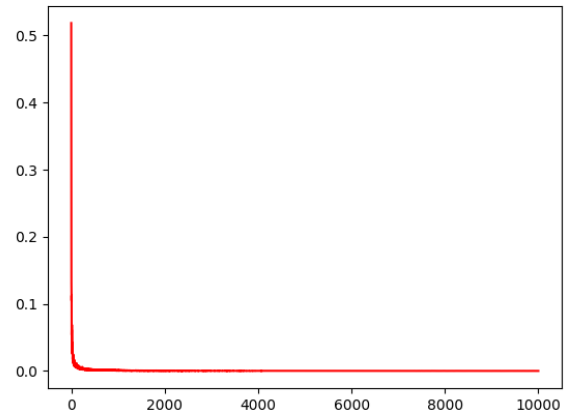

Рис. 3.10: MAE для исходного сэмплирования

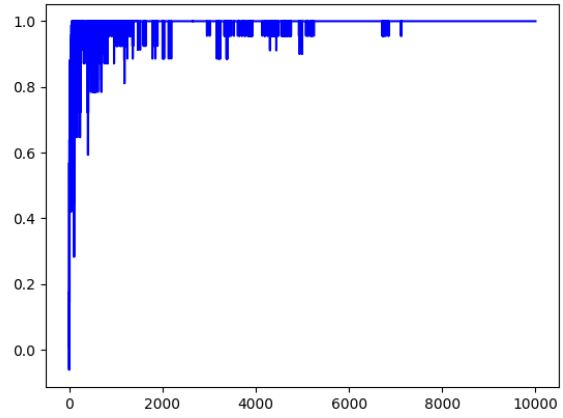

Рис. 3.11: Ранг Спирмена для оригинальноого сэмплирования

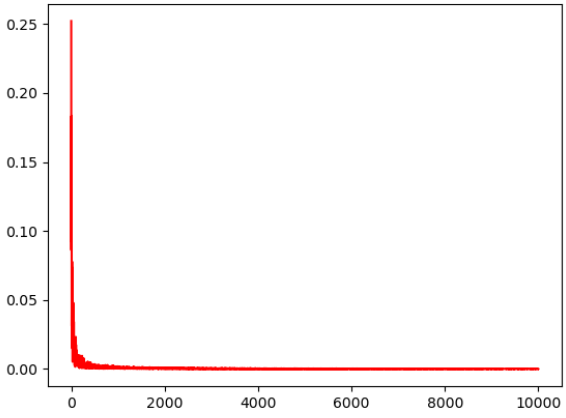

Рис. 3.12: MAE для сэмплирования графа

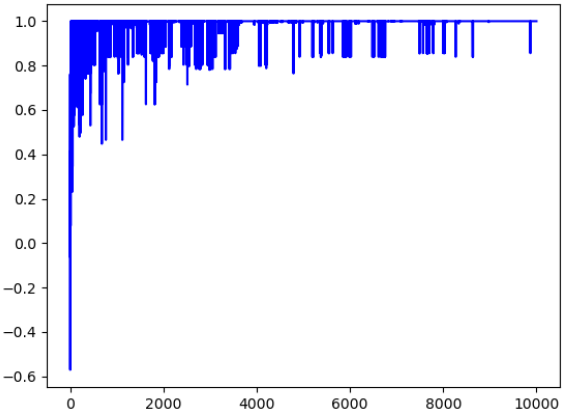

Рис. 3.13: Ранг Спирмена для сэмплирования графа

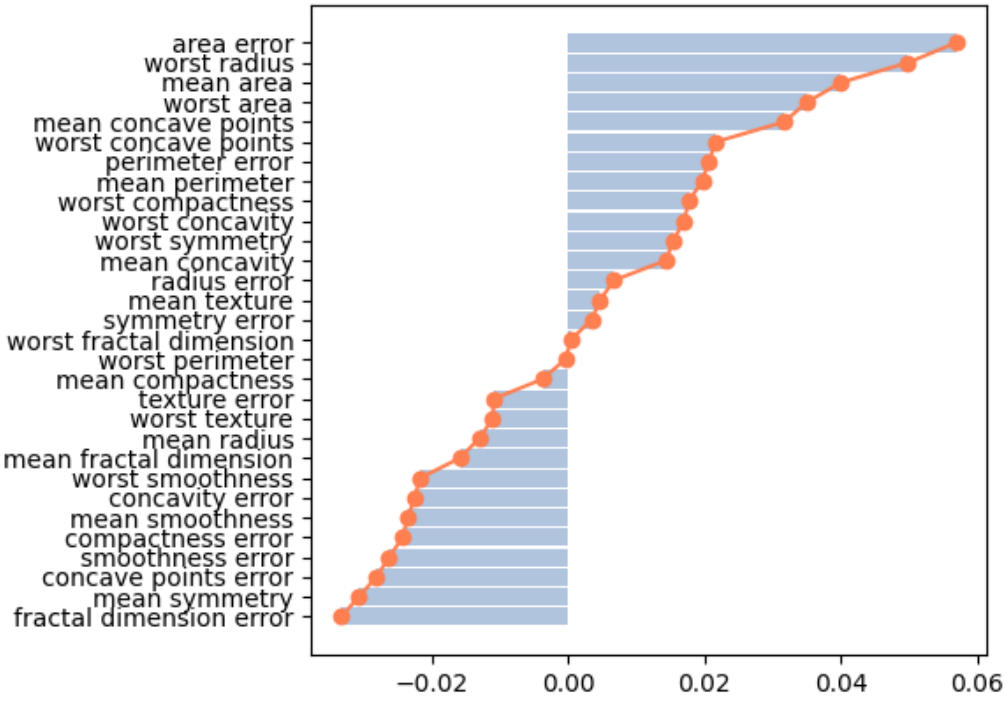

Рис. 3.14: Результат оригинального сэмплирования

Объяснение результатов как оригинального метода сэмплирования Шепли, так и метода сэмплирования Шепли на основе графа представлено на рисунках 3.14 и 3.15. Несмотря на небольшие различия в значениях Шепли и ранжирования, мы по-прежнему можем наблюдать, что 3 из 5 лучших, 8 из 10 лучших и 3 из последних 5 признаков перекрываются, с незначительными расхождениями между ними.

Как видно из таблицы 3.1, первые 10 и последние 10 признаков, полученные методом сэмплирования Шепли на основе графов, дают аналогичные выводы с сопоставимыми вкладами как в первые 10, так и в последние 10 признаков.

Для признаков Mid 10, которые представляют участников среднего уровня, результаты отличаются. Это расхождение возникает из-за границы между положительными и отрицательными вкладами, которая демонстрирует более высокую разницу в значениях Шепли. Проблема может быть связана с отсутствием связей между слабыми взаимодействиями в графе. Однако основанный на графе метод сэмплирования Шепли дает почти идентичные выводы для положительных и отрицательных участников по сравнению с исходным методом сэмплирования, точность для отрицательных и положительных участников до-

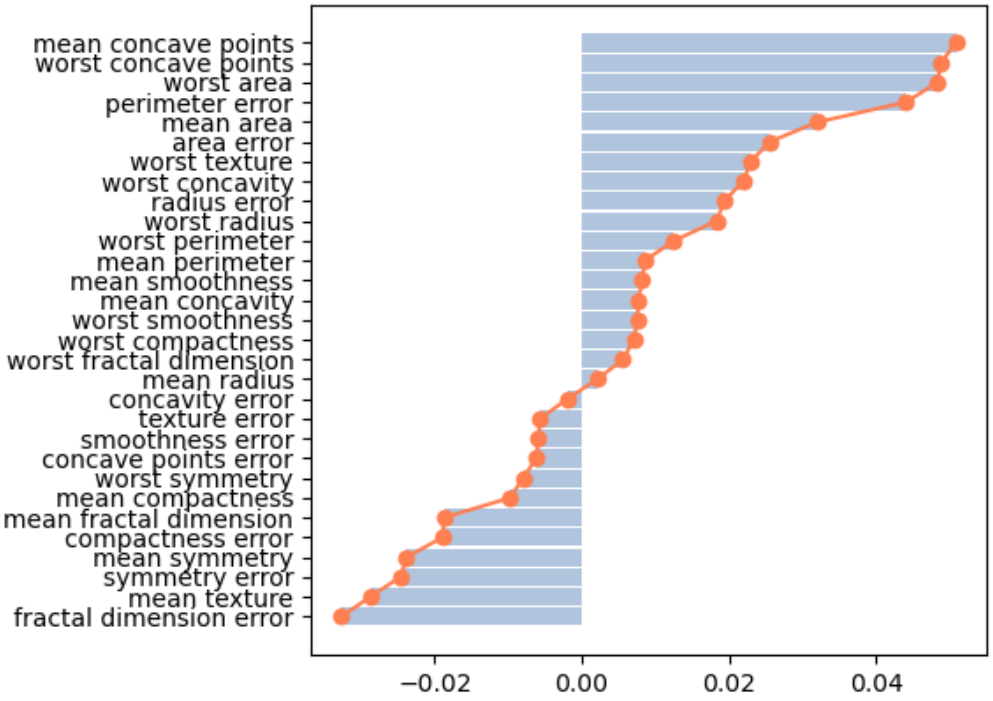

Рис. 3.15: Результат сэмплирования графа

TOP10		MID10		BOT10	
Rank1	Rank2	Rank1	Rank2	Rank1	Rank2
14	8	29	23	1	15
21	28	7	3	10	18
4	24	11	5	25	29
24	13	2	7	17	6
8	4	19	25	5	10
28	14	30	26	16	16
13	22	23	30	15	9
3	27	6	1	18	19
26	11	12	17	9	2
27	21	22	12	20	20

Таблица 3.1: Результат сравнения исходного метода сэмплирования Шепли и графического метода сэмплирования Шепли в разных интервалах

стигает 80%, как показанно в таблица 3.2.

Pos.		Neg.	
Rank1	Rank2	Rank1	Rank2
14	8	6	1
21	28	12	17
4	24	22	12
24	13	1	15
8	4	10	18
28	14	25	29
13	22	17	6
3	27	5	10
26	11	16	16
27	21	15	9
29	23	18	19
7	3	9	2
11	5	20	20
2	7		
19	25		
30	26		
23	30		

Таблица 3.2: Результат сравнения исходного метода сэмплирования Шепли и графического метода сэмплирования Шепли для обоих поз. и отр. участник

3.4 Заключение к главе 3

В этой главе мы предложили метод сэмплирования Шепли, основанный на смещенных графах, для многомерных сложных систем ИИ. В отличие от исходного алгоритма сэмплирования Шепли, наш подход создает карту взаимосвязей на основе коэффициента корреляции Пирсона и использует метод поиска смещенного случайного пути для создания коалиций более высокого качества в качестве сэмплирования. Контролируя размер коалиций (вычислительную сложность), мы сокращаем время расчета E2E, сохраняя при этом точность.

Экспериментальные результаты показывают, что, хотя существуют небольшие различия в признаках ранжирования и значениях Шепли, общие суждения о положительных и отрицательных участниках остаются согласованными между исходным методом и предложенным нами. Более того, общее вычислительное время нашего метода сокращается на 40% по сравнению с исходным алгоритмом при том же количестве итераций.

Кроме того, мы внедрили комбинированное использование корреляции MAE и индекса Спирмена для измерения результатов. Эти методы позволяют нам всесторонне оценить эффективность метода сэмплирования Шепли.

Выводы

В этой диссертации было проведено исследование интерпретируемых методов для многомерных систем ИИ. В ходе исследования были предложены и улучшены два интерпретируемых подхода для многомерных задач ИИ в двух основных приложениях ИИ: системах обнаружения аномалий в логах и системах обнаружения рака. В главе 1 рассматривались системы обнаружения аномалий в логах, и было предоставлено объяснение взаимосвязей между входом и выходом с использованием значения Шепли, основанного на деревьях решений. Кроме того, был разработан метод быстрого вычисления значений Шепли с применением двухуровневого подхода к алгоритму DeepLog на основе нейронных сетей. В главе 2 для систем обнаружения рака двухуровневый подход был улучшен до иерархического, основанного на методе k-means с ограничениями и методе сэмплирования для быстрого вычисления значений Шепли. Результаты двух подходов были схожи, были проведены симуляционные тесты как на локальную, так и на глобальную интерпретируемость. В главе 3 был предложен улучшенный вариант сэмплирования, а также было введено понятие карты отношений. Для построения матрицы взаимосвязей был использован коэффициент корреляции Пирсона. Также для улучшения качества и вычислительной эффективности сэмплирования был предложен подход к сэмплированию на основе смещенного случайного блуждания. Кроме того, были предложены более полная метрика сходимости сэмплирования, объединившая индекс корреляции Спирмена и MAE, и более разумная метрика оценки качества сэмплирования, основанная на сходстве между положительными и отрицательными вкладами.

Основные результаты:

- Были разработаны интерпретируемые подходы на основе значения Шепли для систем обнаружения аномалий в логах и двухуровневое решение, основанное на знаниях предметной области, для объяснения вклада событий в обнаружение аномалий логов. Двухуровневый подход значительно

повысил эффективность вычислений.

- Двухуровневый подход для систем обнаружения рака был улучшен до иерархического двухуровневого подхода, основанного на кластеризации k-means с ограничениями, и использовании сэмплирования для быстрого вычисления значений Шепли. Результаты двух подходов были схожи, были проведены симуляционные тесты как на локальную, так и на глобальную интерпретируемость, которые это подтвердили.

- Для дальнейшего улучшения сэмплирования была предложена концепция карты взаимосвязей с использованием коэффициента корреляции Пирсона, а также применение смещенного случайного блуждания для улучшения качества сэмплирования и вычислительной эффективности. Кроме того, были предложены более комплексные показатели качества и сходимости сэмплирования.

Литература

[1] Lundberg SM et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature Biomed. Engin., 2.10 (2018): 749-760.

[2] Holzinger A et al. What do we need to build explainable AI systems for the medical domain? arXiv preprint, arXiv:1712.09923. – 2017.

[3] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Computing Surveys, 2009, 41(3).

[4] Hawkins D. Identification of outliers. Springer Netherlands, 1980, P. 188.

[5] Chalapathy R, Chawla S. Deep learning for anomaly detection: a survey, arXiv: Learning, 2019.

[6] Tkachenko R, Izonin I. Model and principles for the implementation of neural-like structures based on geometric data transformations. Adv Intell Syst Comput 754: 578–587.

[7] Izonin I, Tkachenko R, Kryvinska N, Tkachenko P. Multiple linear regression based on coefficients identification using non-iterative SGTM Neural-Like Structure. In International Work-Conference on Artificial Neural Networks, Springer, Cham, 2019 June, pp: 467-497.

[8] Tkachenko R, Izonin I, Vitynskyi P, Lotoshynska N, Pavlyuk O. Development of the noniterative supervised learning predictor based on the ITO decomposition and SGTM neural-like structure for managing medical insurance costs. Data, 2018, 3(4), 46.

[9] Antwarg L, Shapira B. Explaining anomalies detected by autoencoders using SHAP. arXiv preprint, arXiv:1903.02407. – 2019.

[10] Shapley LS. (August 21, 1951). Notes on the n-Person Game – II: The Value of an n-Person Game. Santa Monica, Calif.: RAND Corporation.

[11] Leon A P, Nikolay A Z. Game Theory (2nd Edition), World Scientific, 2016.

[12] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Neural Inform. Processing Syst., 2017, pp. 4765–4774.

[13] Sundararajan M, Najmi A. The many shapley values for model explanation. arXiv preprint, arXiv: 1908.08474, 2019.

[14] Aas K, Jullum M, Løland A. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. arXiv preprint, arXiv:1903.10464, 2019.

[15] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee SI. From local explanations to global understanding with explainable AI for trees. Nature machine intelligence, 2020, 2(1): 2522–5839.

[16] Vega Garcıa M, Aznarte JL. Shapley additive explanations for NO2 forecasting. Ecol. Inform., Mar 2020, 56: 101039.

[17] Ribeiro MT, Singh S, Guestrin C. Why should I trust you? Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, – 2016. – . 1135-1144.

[18] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences //arXiv preprint arXiv:1704.02685. – 2017.

[19] Montavon G et al. Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning. – Springer, Cham, 2019. – . 193-209.

[20] Arun D, Paul R. Opportunities and challenges in explainable artificial intelligence (XAI): a survey, arXiv, 2020.

[21] Chen M, Zheng AX, Lloyd J, Jordan MI, Brewer E. Failure diagnosis using decision trees. International Conference on Autonomic Computing, 2004. Proceedings., New York, NY, USA, 2004, pp. 36-43, doi: 10.1109/ICAC.2004.1301345.

[22] Liang YL, Zhang YY, Xiong H, Sahoo R. Failure Prediction in IBM BlueGene/L Event Logs.

[23] Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, 1285-1298.

[24] Xu W, Huang L, Fox A, Patterson D, Jordan MI. Large-Scale System Problems Detection by Mining Console Logs.

[25] Nikolay VK. Strategic stability of coalitions technological alliance parameters: a two-level cooperation. Contributions to Game Theory and Management, 2015, Volume 8, 111–136

[26] He SL, Zhu JM, He PJ, Michael RL. Experience report: system log analysis for anomaly detection. IEEE International Symposium on Software Reliability Engineering (ISSRE), 2016.

[27] Shapley L. A value for n-person games. Contributions to the Theory of Games. 1953. 2(1): 307–317.

[28] Mann I, Shapley LS. Values of large games 6: Evaluating the electoral college exactly. Tech. Rep., Rand Corp Santa Monica CA (1962).

[29] Castro J, G´omez D, Tejada J. Polynomial calculation of the shapley value based on sampling. Comput. Oper. Res. 2009, 36: 1726–1730.

[30] Maleki S, Tran-Thanh L, Hines G, Rahwan T, Rogers A. Bounding the estimation error of sampling based shapley value approximation. arXiv preprint, arXiv, 2013, 1306.4265.

[31] Adadi A, Berrada M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)[J]. IEEE Access, 2018, 6: 52138-52160.

[32] Balakrishnama S, Ganapathiraju A. Linear discriminant analysis-a brief tutorial[C]//Institute for Signal and information Processing. 1998, 18(1998): 1-8.

[33] Bradley, P. S., K. P. Bennett, and Ayhan Demiriz. "Constrained k-means clustering."Microsoft Research, Redmond (2000): 1-8.

[34] Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial–temporal data[J]. Data and knowledge engineering, 2007, 60(1): 208-221.

[35] Cath C, Wachter S, Mittelstadt B, et al. Artificial intelligence and the 'good society': the US, EU, and UK approach[J]. Science and engineering ethics, 2018, 24(2): 505-528.

[36] Challen R, Denny J, Pitt M, et al. Artificial intelligence, bias and clinical safety[J]. BMJ Quality and Safety, 2019, 28(3): 231-237.

[37] Cheadle C, Vawter M P, Freed W J, et al. Analysis of microarray data using Z score transformation[J]. The Journal of molecular diagnostics, 2003, 5(2): 73-81.

[38] Fatima S S, Wooldridge M, Jennings N R. An analysis of the shapley value and its uncertainty for the voting game[M]//Agent-Mediated Electronic Commerce. Designing Trading Agents and Mechanisms. Springer, Berlin, Heidelberg, 2005: 85-98.

[39] Kuswanto H, Mubarok R, Ohwada H. Classification Using Naive Bayes to Predict Radiation Protection in Cancer Drug Discovery: a Case of Mixture Based Grouped Data. International Journal of Artificial Intelligence, 2019, 17(1), 186-203.

[40] Liu F T, Ting K M, Zhou Z H. Isolation forest[C]//2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008: 413-422.

[41] Liu F T, Ting K M, Zhou Z H. Isolation-based anomaly detection[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 6(1): 1-39.

[42] Lui A, Lamb G W. Artificial intelligence and augmented intelligence collaboration: regaining trust and confidence in the financial sector[J]. Information and Communications Technology Law, 2018, 27(3): 267-283.

[43] Patcha A, Park J M. An overview of anomaly detection techniques: Existing solutions and latest technological trends[J]. Computer networks, 2007, 51(12): 3448-3470.

[44] Petrosian O, Barabanov A. Looking Forward Approach in cooperative differential games with uncertain stochastic dynamics[J]. Journal of Optimization Theory and Applications, 2017, 172(1): 328-347.

[45] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding[J]. science, 2000, (290): 2323-2326.

[46] Tomin N, Zhukov A, Sidorov D, Kurbatsky V, Panasetsky D, Spiryaev V. Random forest based model for preventing large-scale emergencies in power systems. International Journal of Artificial Intelligence, 2015, 13(1), 211-228.

[47] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37-52.

[48] Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI."Information fusion 58 (2020): 82-115.

[49] Guidotti, Riccardo, et al. "A survey of methods for explaining black box models."ACM computing surveys (CSUR) 51.5 (2018): 1-42.

[50] Štrumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions."Knowledge and information systems 41 (2014): 647-665. APA

[51] Zhang, Yuyi, et al. "XAI evaluation: evaluating black-box model explanations for prediction."2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT). IEEE, 2021.

[52] Dikshit, Abhirup, and Biswajeet Pradhan. "Interpretable and explainable AI (XAI) model for spatial drought prediction."Science of the Total Environment 801 (2021): 149797.

[53] Roshan K, Zafar A. Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation (SHAP)[J]. arXiv preprint arXiv:2112.08442, 2021.

[54] Jinying Zou, Ovanes Petrosian. "Explainable AI: Using Shapley value to explain complex anomaly detection ML-based systems."Machine learning and artificial intelligence 332 (2020): 152.

[55] Walia S, Kumar K, Agarwal S, et al. Using XAI for Deep Learning-Based Image Manipulation Detection with Shapley Additive Explanation[J]. Symmetry, 2022, 14(8): 1611.

[56] Granot, Daniel, Jeroen Kuipers, and Sunil Chopra. "Cost allocation for a tree network with heterogeneous customers."Mathematics of Operations Research 27.4 (2002): 647-661.

[57] Chalkiadakis, Georgios, Edith Elkind, and Michael Wooldridge. "Computational aspects of cooperative game theory."Synthesis Lectures on Artificial Intelligence and Machine Learning 5.6 (2011): 1-168.

[58] Castro, Javier, et al. "Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation."Computers & Operations Research 82 (2017): 180-188.

[59] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning Important Features Through Propagating Activation Differences”. In: arXiv preprint arXiv:1704.02685 (2017).

[60] Zou, J., et al. "High-dimensional explainable AI for cancer detection."International Journal of Artificial Intelligence 19.2 (2021): 195.

[61] Pearson, Karl. "VII. Note on regression and inheritance in the case of two parents."proceedings of the royal society of London 58.347-352 (1895): 240-242.

[62] Hanley, Anthony JG, et al. "Prediction of type 2 diabetes using simple measures of insulin resistance: combined results from the San Antonio Heart Study, the Mexico City Diabetes Study, and the Insulin Resistance Atherosclerosis Study."Diabetes 52.2 (2003): 463-469.

[63] Oldham, M. C., Horvath, S., Geschwind, D. H. (2006). Conservation and evolution of gene coexpression networks in human and chimpanzee brains. Proceedings of the National Academy of Sciences, 103(47), 17973-17978.

[64] Spearman, Charles. "The proof and measurement of association between two things."The American journal of psychology 100.3/4 (1987): 441-471.